Efficient Interpolation in Wu List Decoding Algorithm

Peter Trifonov, *Member, IEEE,* and Moon Ho Lee, *Senior Member, IEEE*

***Abstract*—The interpolation step of Wu list decoding algorithm for Reed-Solomon codes is considered. The problem is reformulated as construction of a partially homogenized interpolation polynomial. A generalization of the binary interpolation algorithm, which is based on the novel formulation of the interpolation step, is provided. It enables complexity reducion both with respect to the Wu method based on the Iterative Interpolation Algorithm (IIA), as well as the Guruswami-Sudan method based on re-encoding and the binary interpolation algorithm.**

I. Introduction

Reed-Solomon codes are extensively used in modern communication and storage systems. Classical algebraic decoding algorithms are able to correct up to $(d-1)/2$ errors, where d is the minimum distance of the code. List decoding can significantly increase the error correction radius at the expense of possible non-uniqueness of the decoder output. Guruswami and Sudan have proposed a polynomial-time decoding algorithm for Reed-Solomon codes [1]. However, its complexity remains too high for practical applications despite of numerous complexity reduction methods proposed recently [2], [3], [4], [5]. Wu proposed to use rational curve fitting to derive the solutions of the list decoding problem from the output of the classical Berlekamp-Massey algorithm [6]. This approach requires much smaller root multiplicity, which automatically results in smaller complexity. However, the complexity still remains much higher than for the case of classical algorithms.

In this paper a novel derivation of the Wu list decoding method is given. The new formulation of the interpolation step avoids roots at infinity, which are used in the description of the original method [6]. This allows one to introduce the ideal of interpolation polynomials, enabling thus application of the fast binary interpolation algorithm, which was introduced in [5] for the case of Guruswami-Sudan algorithm.

The paper is organized as follows. The new derivation of the Wu method is given in Section III. The rational curve fitting problem, which is used in the considered method, is treated in Section IV. Section V presents a generalization of the binary interpolation method to the case of rational curve fitting problem. Numeric results are provided in Section VI. Finally, some conclusions are drawn.

II. Notation

- $[Q_i, 0\leq i\leq v]=\left\{\sum_{i=0}^{v} p_i(x)Q_i | p_i(x)\in\mathbb{F}[x]\right\}$ is the module generated by multivariate polynomials Q_i with coefficients in $\mathbb{F}$.
- $\operatorname{LT} Q$ is the leading term of the polynomial Q with respect to some term ordering.
- $\operatorname{ydeg} Q=j$ iff $\operatorname{LT} Q(x,y)=ax^uy^j$ (in the case of bivariate polynomials) or $\operatorname{LT} Q(x,y,z)=ax^uy^jz^{\rho-j}$ (in the case of trivariate partially homogenized polynomials with some fixed ρ) for some $a\in\mathbb{F}$ and $u\in\mathbb{Z}$.
- $\operatorname{xdeg} Q(x,y,z)=u$ iff $\operatorname{LT} Q(x,y,z)=ax^uy^jz^{\rho-j}$ for some $a\in\mathbb{F}$ and $u\in\mathbb{Z}$.
- $\Delta(\mathcal{B})=\sum_{j=0}^{s}\operatorname{xdeg}\mathcal{B}_j$, where $\mathcal{B}=(B_0(x,y,z),\ldots,B_s(x,y,z))$.
- $\operatorname{wdeg}_{(a,b,c)}\alpha x^uy^vz^w=au+bv+cw, \alpha\in\mathbb{F}\setminus\{0\}$, is the (a,b,c)-weighted degree of monomial $\alpha x^uy^vz^w$. The (a,b,c)-weighted degree of a polynomial is equal to the highest weighted degree of its non-zero monomials.

III. A simple derivation of Wu algorithm

$(n,k,n-k+1)$ Reed-Solomon code is defined as a set of vectors $(f(x_1),\ldots,f(x_n))$, where $f(x)=\sum_{i=0}^{k-1}f_ix^i, f_i\in\mathbb{F}$, and $x_i\in\mathbb{F}$ are distinct code locators. Let $(y_1,\ldots,y_n)\in\mathbb{F}^n$ be some codeword corrupted by channel noise. The list decoding problem consists in finding all pairs $(f(x),\sigma(x))$, such that $\deg f(x)<k$ and $\sigma(x_i)=0$ for at most t distinct $x_i : f(x_i)\neq y_i$, so that $y_i\sigma(x_i)=f(x_i)\sigma(x_i), i=1..n$. Here t is the decoding radius, $f(x)$ identifies the corresponding codeword, and $\sigma(x)$ is the error locator polynomial. Observe that one can recover $\sigma(x)$ from $f(x)$ and vice versa.

Any such pair of polynomials $f(x),\sigma(x)$ can be represented as a bivariate polynomial

$$Q(x,y)=y\sigma(x)-f(x)\sigma(x). \tag{1}$$

It can be seen that it has n roots (x_i,y_i). Hence, it belongs to the module $\mathcal{M}=[\phi(x),y-T(x)]$, where $\phi(x)=\prod_{i=1}^{n}(x-x_i)$, and $T(x)$ is a polynomial[1] such that $T(x_i)=y_i$. Let the polynomials $Q'(x,y)=q_{00}(x)+yq_{10}(x)$ and $Q''(x,y)=$

Manuscript received October 19, 2011; revised March 16, 2012; accepted May 29, 2012. This work was supported in part by Russian Ministry of Education and Science under Contract 07.514.11.4101, in part by the Korean World Class University (WCU) Project WCUR32-2012-000-20014-0, and in part by the Korean National Research Foundation under Projects FR2010-0020942 and MEST2012-002521. This paper was presented in part at the IEEE R8 International Conference on Computational Technologies in Electrical and Electronics Engineering 2010.

P. Trifonov is with the Distributed Computing and Networking Department, Saint-Petersburg State Polytechnic University, Polytechnicheskaya str., 21, office 104, 194021, Saint-Petersburg, Russia (e-mail: petert@dcn.ftk.spbstu.ru).

M. H. Lee is with the Division of Electronics & Information Engineering, Chonbuk National University, 664-14 1GA Dekjin-Dong, Jeonju City, Jeonbuk, South Korea 561-756 (e-mail:moonho@chonbuk.ac.kr)

[1] Observe that given an arbitrary $T(x)$ satisfying these constraints one can obtain the smallest degree interpolation polynomial as $\tilde{T}(x)\equiv T(x) \bmod \phi(x)$. Since $\phi(x)\in\mathcal{M}$, one obtains $\mathcal{M}=[\phi(x),y-T(x)]=[\phi(x),y-\tilde{T}(x)]$.

$q_{01}(x) + yq_{11}(x)$ be another basis of this module. Then any $Q(x, y) \in \mathcal{M}$ can be represented as

$$Q(x, y) = a(x)Q'(x, y) + b(x)Q''(x, y). \qquad (2)$$

Assume now that $Q'(x, y)$ and $Q''(x, y)$ constitute a Gröbner basis of $\mathcal{M}$ with respect to $(1, k-1)$-weighted degree lexicographic ordering with $y \prec x$, so that $\operatorname{ydeg} Q'(x, y) = 0$ and $\operatorname{ydeg} Q''(x, y) = 1$. By the properties of Gröbner basis one obtains that any valid $Q(x, y)$ can be represented as in (2), where $\operatorname{LT} Q(x, y) = \operatorname{LT} a(x) \operatorname{LT} Q'(x, y)$ or $\operatorname{LT} Q(x, y) = \operatorname{LT} b(x) \operatorname{LT} Q''(x, y)$, and the polynomials $a(x)$ and $b(x)$ satisfying (2) can be recovered via the multivariate division algorithm [7]. This implies that $\deg b(x) \leq w_2 = t - \deg q_{11}(x)$, and $\deg a(x) \leq w_1 = t + k - 1 - \deg q_{00}(x)$. Hence, list decoding of a Reed-Solomon code can be implemented by enumerating all such polynomials $a(x), b(x)$, constructing the corresponding bivariate polynomial $Q(x, y)$ and checking if it can be factored as in (1), subject to the following constraints:

1) $\sigma(x)$ must be a valid error locator polynomial.
2) $f(x)$ must agree with $(y_1, \ldots, y_n)$ in sufficiently many positions.

That is, the algorithm involves the following steps:

1) Construct $T(x) : T(x_i) = y_i, i = 1..n$.
2) Find polynomials $Q'(x, y) = q_{00}(x) + yq_{10}(x)$ and $Q''(x, y) = q_{01}(x) + yq_{11}(x)$ being a Gröbner basis of the module $\mathcal{M} = [\phi(x), y - T(x)]$ with respect to $(1, k-1)$-weighted degree lexicographic ordering. This step is similar to the extended Euclidean algorithm with an early termination condition, as used in Gao decoding method [8].
3) [Rational curve fitting] Find all pairs of coprime polynomials $a(x), b(x) : \deg a(x) \leq w_1 = t + k - 1 - \deg q_{00}(x), \deg b(x) \leq w_2 = t - \deg q_{11}(x)$, such that
$$\sigma(x) = a(x)q_{10}(x) + b(x)q_{11}(x) \qquad (3)$$
has at most t roots.
4) For each j reconstruct the codeword from symbols y_i such that x_i are not roots of $\sigma(x)$.

The described algorithm can be considered as a frequency-domain interpretation of the Wu method [6], and has been independently derived in [9].

Recall, that the original Wu method is based on the analytical continuation of the Berlekamp-Massey algorithm, and consists in finding all pairs of polynomials $(a(x), b(x))$, such that the polynomial

$$\Lambda^*(x) = a(x)\Lambda(x) + xB(x)b(x) \qquad (4)$$

has at most t distinct roots α^{-j_i}, where $j_i, 1 \leq i \leq t$, is the position of the i-th error, and $\Lambda(x)$ and $B(x)$ are the polynomials obtained by the Berlekamp-Massey algorithm from the standard syndrome vector. Application of the Gröbner basis language makes the derivation of the algorithm much simpler.

IV. Rational curve fitting

A. Wu method

The main idea of the Wu method is to avoid exhaustive search for polynomials $a(x), b(x)$. Instead, the appropriate polynomials can be identified algebraically in a way similar to the Guruswami-Sudan algorithm. More specifically, given the polynomials $\Lambda(x)$ and $B(x)$ (see (4)), the appropriate pairs of polynomials $a(x), b(x)$ can be identified by finding a polynomial $S(x, y)$ having roots $(x_i, -\frac{\Lambda(x_i^{-1})}{x_i^{-1}B(x_i^{-1})})$ of multiplicity r for some $r \geq 1$, and solving the equation $a^\rho(x)S(x, b(x)/a(x)) = 0$, where ρ is the degree of $S(x, y)$ with respect to variable y. It may happen that $B(x_i^{-1}) = 0$ for some i, i.e. some interpolation points may have infinity as the second component. By definition, $S(x, y)$ has root (x_i, ∞) iff $y^{\deg_y S(x,y)} S(x, y^{-1})$ has root $(x_i, 0)$. However, the existing bivariate interpolation algorithms [2], [10] cannot be immediately used to obtain $S(x, y)$, since most of them construct a basis of the ideal of polynomials with prescribed roots, and the described set is not an ideal.

Indeed, the polynomials $1 - xy$ and $1 - x^2y$ have a root $(0, \infty)$. However, the polynomial $x(1-xy)-(1-x^2y) = x-1$ does not have it.

The same problem arises in the reformulated algorithm based on (3).

B. Interpolation by partially homogeneous polynomials

The above described technical difficulty can be avoided by introducing partially homogenized polynomials $S(x, y, z) = \sum_{j=0}^{\rho} s_j(x)z^{\rho-j}y^j$. The following lemma establishes the relationship between the proposed approach and the original Wu method.

Lemma 1. *Let $S(x, y, z) = \sum_{j=0}^{\rho} \sum_i s_{ji}x^iy^jz^{\rho-j}$ be a polynomial homogeneous in variables y and z. It has roots of multiplicity r at points $(x_0, \alpha y_0, -\alpha z_0)$ for any $\alpha \in \mathbb{F}$, where y_0 and z_0 are not simultaneously zero, if and only if*

- *$\hat{S}(x, \theta) = \sum_{j=0}^{\rho} \sum_i s_{ji}x^i\theta^j$ has a root $(x_0, -y_0/z_0)$ of multiplicity r (for $z_0 \neq 0$);*
- *$\tilde{S}(x, \theta) = \sum_{j=0}^{\rho} \sum_i s_{\rho-j,i}x^i\theta^j$ has a root $(x_0, -z_0/y_0)$ of multiplicity r (for $y_0 \neq 0$).*

Proof: Assume without loss of generality that $z_0 \neq 0$. $S(x, y, z) = \sum_{j=0}^{\rho} \sum_{i \geq 0} s_{ji}x^iy^jz^{\rho-j}$ has roots of multiplicity r at points $(x_0, \alpha y_0, -\alpha z_0)$ if and only if its Hasse derivatives at these points of total order less than r are equal to zero, i.e.

$$\sum_{i' \geq u} \sum_{j'=v}^{\rho-w} \binom{i'}{u} \binom{j'}{v} \binom{\rho - j'}{w} s_{j'i'} x_0^{i'-u} \frac{(\alpha y_0)^{j'-v}}{(-\alpha z_0)^{j'+w-\rho}} = 0$$

for all $u, v, w \geq 0$, s.t. $u + v + w < r$. Then for $w = 0$ one obtains

$$(-z_0)^{\rho-v} \sum_{i' \geq u} \sum_{j'=v}^{\rho} \binom{i'}{u} \binom{j'}{v} s_{j'i'} x_0^{i'-u} \left(-\frac{y_0}{z_0}\right)^{j'-v} = 0$$

for all $u, v : u + v < r$, i.e. $(x_0, -y_0/z_0)$ is a root of multiplicity r of $\hat{S}(x, \theta) = \sum_{j=0}^{\rho} \sum_{i \geq 0} s_{ji}x^i\theta^j$.

If $(x_0, -y_0/z_0)$ is a root $\hat{S}(x,\theta)$ of multiplicity r, then

$$\hat{S}(x,\theta) = \sum_{\substack{u+v\geq r\\ v\leq\rho}} s^{[u,v]}(x-x_0)^u(\theta+y_0/z_0)^v.$$

Hence,

$$\begin{aligned} &S(x,y,z) = z^\rho \hat{S}(x,y/z)\\ &= \sum_{\substack{u+v\geq r\\ v\leq\rho}} \frac{s^{[u,v]}}{(-z_0)^v}(x-x_0)^u(yz_0+zy_0)^v z^{\rho-v}\\ &= \sum_{\substack{u+v\geq r\\ v\leq\rho}} \frac{s^{[u,v]}}{(-z_0)^v}(x-x_0)^u((y-\alpha y_0)z_0+(z+\alpha z_0)y_0)^v z^{\rho-v}. \end{aligned}$$

It can be seen that the polynomial $S(x+x_0, y+\alpha y_0, z-\alpha z_0)$ does not have any terms of total degree less than r for any α, so the points $(x_0, \alpha y_0, -\alpha z_0)$ are its roots of multiplicity r. ■

The following are reformulations of [1, Lemma 4,5].

Lemma 2. *Let $S(x,y,z) = \sum_{j=0}^{\rho} s_j(x)y^j z^{\rho-j}$ be a polynomial having root of multiplicity r at points $(x_0, \alpha y_0, -\alpha z_0)$ for any α, where y_0 and z_0 are not simultaneously zero. If $a(x), b(x)$ are polynomials such that $z_0 a(x_0) + y_0 b(x_0) = 0$, then $(x-x_0)^r | S(x, a(x), b(x))$.*

Proof: Assume w.l.o.g. that $z_0 \neq 0$. Then by Lemma 1 one obtains that $\widehat{S}(x,\theta) = \sum_{j=0}^{\rho} s_j(x)\theta^j$ has a root $(x_0, -y_0/z_0)$ of multiplicity r, i.e.

$$\widehat{S}(x,\theta) = \sum_{\substack{u+v\geq r\\ v\leq\rho}} \widehat{s}^{[u,v]}(x-x_0)^u(\theta+y_0/z_0)^v.$$

Hence,

$$S(x,y,z) = \sum_{\substack{u+v\geq r\\ v\leq\rho}} \frac{\widehat{s}^{[u,v]}}{z_0^v}(x-x_0)^u(yz_0+zy_0)^v z^{\rho-v}.$$

The statement of the lemma follows from the fact that x_0 is a root of polynomial $z_0 a(x) + y_0 b(x)$. ■

Lemma 3. *Let $S(x,y,z) = \sum_{j=0}^{\rho} s_j(x)y^j z^{\rho-j}$ be a polynomial such that $\mathrm{wdeg}_{(1,w_1,w_2)} S(x,y,z) < rt$, and points $(x_i, \alpha y_i, -\alpha z_i), i = 1..n$ are its roots of multiplicity r for any α, where y_i and z_i are not simultaneously zero. If $a(x)$ and $b(x)$ are the polynomials such that $\deg a(x) \leq w_1$, $\deg b(x) \leq w_2$ and $z_i a(x_i) + y_i b(x_i) = 0$ for at least t points (x_i, y_i, z_i), then $S(x, a(x), b(x)) = 0$.*

Proof: By Lemma 2, for any such point $(x-x_i)^r | S(x, a(x), b(x))$. The degree of $g(x) = S(x, a(x), b(x))$ is at most $\mathrm{wdeg}_{(1,w_1,w_2)} S(x,y,z) < rt$. Hence, the only possibility for a polynomial $\prod_i (x-x_i)^r$ of degree rt to be a divisor of $g(x)$ is $S(x, a(x), b(x)) = 0$. ■

The root multiplicity constraints give $nr(r+1)/2$ linear equations. It is possible to solve this system of equations and obtain the required polynomial if the number of unknowns in it exceeds the number of equations, i.e. $\sum_{j=0}^{\rho}(rt - jw_1 - (\rho - j)w_2) = rt(\rho+1) - w\frac{\rho(\rho+1)}{2} > n\frac{r(r+1)}{2}$, where $w = w_1 + w_2$. For $w = 0$ this implies $r = 1$, and

$$\rho > n\frac{r+1}{2t} - 1 = \frac{n}{t} - 1. \tag{5}$$

For $w > 0$ one obtains

$$\rho_l = \frac{2rt - w - \sqrt{D}}{2w} < \rho < \frac{2rt - w + \sqrt{D}}{2w} = \rho_h, \tag{6}$$

where $D = (w+2rt)^2 - 4wnr(r+1) > 0$. The latter inequality implies

$$r > \frac{\left(n - t + \sqrt{n^2 - 2tn + wn}\right) w}{2(t^2 - wn)}. \tag{7}$$

This can be satisfied if $t^2 - wn \geq 0$. Since any Gröbner basis of $\mathcal{M}$ satisfies $\deg q_{11}(x) + \deg q_{00}(x) = n$ [5], one obtains $w = w_1 + w_2 = 2t + (k-1) - n$. Hence, decoding is possible if $t < n - \sqrt{n(k-1)}$.

In general, r should be selected as small as possible in order to minimize the decoding complexity. However, for small r it may happen that the range given by (6) does not include any integer numbers. In this case one has to increase r until suitable ρ is found. In order to guarantee the existence of integer ρ one has to ensure that $\rho_h - \rho_l = \sqrt{D}/w > 1$. Simple calculation results in

$$r > \frac{w(n-t)}{t^2 - wn}.$$

Theorem 1. *Let $\overline{y} = (y_1, \ldots, y_n)$ be some vector in $\mathbb{F}^n$. Consider polynomials $Q'(x,y) = q_{00}(x) + yq_{10}(x), Q''(x,y) = q_{01}(x) + yq_{11}(x)$ being a Gröbner basis of module $\mathcal{M} = [\phi(x), y - T(x)]$ with respect to $(1, k-1)$-weighted degree lexicographic ordering, where $T(x) : T(x_i) = y_i, 1 \leq i \leq n, \phi(x) = \prod_{i=1}^{n}(x - x_i)$. If $t < n - \sqrt{n(k-1)}$ and parameters r, ρ satisfy (5)–(7), then all codewords $\overline{c} = (c_1, \ldots, c_n)$ of $(n, k, n-k+1)$ RS code over $\mathbb{F}$ such that $d_H(\overline{c}, \overline{y}) = t' \leq t$, can be identified by polynomials $\sigma(x) = a(x)q_{10}(x) + b(x)q_{11}(x)$, such that $c_i \neq y_i \Leftrightarrow \sigma(x_i) = 0$, $S(x, a(x), b(x)) = 0$. Here $S(x,y,z) = \sum_{i=0}^{\rho} s_i(x)y^i z^{\rho-i}$ is a polynomial, such that for all $\alpha \in \mathbb{F}$ the points $(x_i, \alpha q_{11}(x_i), -\alpha q_{10}(x_i)), 1 \leq i \leq n$, are its roots of multiplicity r, and $\mathrm{wdeg}_{(1,w_1,w_2)} S(x,y,z) < rt$, where $w_1 = t + k - 1 - \deg q_{00}(x), w_2 = t - \deg q_{11}(x)$.*

Proof: The constraints (5)–(7) ensure that the required polynomial $S(x,y,z)$ can be indeed constructed. For any codeword $\overline{c}$ there exist polynomials $f(x), \sigma(x)$ such that $\deg f(x) < k, f(x_i) = c_i, 1 \leq i \leq n, \sigma(x) = \prod_{i: c_i \neq y_i}(x - x_i)$. If $d_H(\overline{c}, \overline{y}) = t' \leq t$, then $\sigma(x) = a(x)q_{10}(x) + b(x)q_{11}(x)$, where $\deg a(x) \leq t' + k - 1 - \deg q_{00}(x)$ and $\deg b(x) \leq t' - \deg q_{11}(x)$. Let $\omega(x)$ be a smallest degree polynomial having $t - t'$ roots from the set of non-erroneous position locators $\{x \in \{x_1, \ldots, x_n\} | c_i = y_i\}$. Then $\sigma'(x) = \sigma(x)\omega(x) = a'(x)q_{10}(x) + b'(x)q_{11}(x)$ has t distinct roots, and $\deg a'(x) \leq t + k - 1 - \deg q_{00}(x)$, $\deg b'(x) \leq t - \deg q_{11}(x)$. Lemma 3 implies that $0 = S(x, a'(x), b'(x)) = S(x, \omega(x)a(x), \omega(x)b(x)) = \omega^\rho(x)\sum_{j=0}^{\rho} s_j(x)a^j(x)b^{\rho-j}(x) = \omega^\rho(x)S(x, a(x), b(x))$, i.e. $S(x, a(x), b(x)) = 0$. ■

Given a polynomial $S(x,y,z)$ satisfying the above constraints, the polynomials $a(x), b(x)$ can be recovered by the modified Roth-Ruckenstein algorithm given in [6]. It appears that ρ gives an upper bound on the list size, i.e. the number of distinct error locator polynomials $\sigma(x)$ which can be obtained in this way.

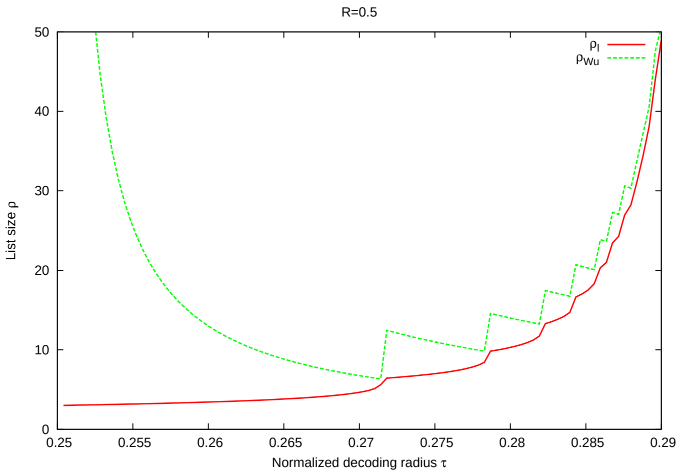

Fig. 1. Comparison of the bounds on list size

In order to analyze the behavior of the obtained bounds, let us introduce the normalized correctable error fraction $\tau=t/n$ and code rate $R=(k-1)/n$. Then (7) can be rewritten as

$$r>\left\lceil\frac{(1-\tau+\sqrt{R})(2\tau+R-1)}{2(\tau^2-2\tau-R+1)}\right\rceil.$$

Similarly,

$$\rho_l=\frac{r\tau-\sqrt{(\tau(r+1)+\frac{R-1}{2})^2-(2\tau+R-1)r(r+1)}}{2\tau+R-1}-\frac{1}{2}.$$

These estimates apply to the original Wu algorithm as well.

Figure 1 presents the comparison of this estimate and the one derived in [6] ($\rho_{Wu}=\left\lfloor\frac{rt}{w}\right\rfloor=\left\lfloor\frac{r\tau}{2\tau+R-1}\right\rfloor$). It can be seen that the new one is substantially better, especially for small values of τ. Furthermore, its behavior is much more natural, i.e. list size increases with decoding radius.

V. Efficient interpolation

A. Partially homogenized polynomials

As it was shown above, all pairs of polynomials $(a(x),b(x))$, such that the polynomial $\sigma(x)=a(x)q_{10}(x)+b(x)q_{11}(x)$ has t distinct roots, are given by the equation $S(x,a(x),b(x))=0$, where $S(x,y,z)$ is a polynomial having roots $(x_i,\alpha q_{11}(x_i),-\alpha q_{10}(x_i))$ of multiplicity r with $(1,w_1,w_2)$-weighted degree less than rt. It can be found in a Gröbner basis of the ideal I_r of polynomials having these roots [11]. However, the full Gröbner basis of this ideal contains a lot of polynomials not satisfying the constraint (6), which are useless for decoding purposes. It is sufficient to consider just a submodule $M_{\rho,r}=\{S(x,y,z)\in I_r|S(x,y,z)=\sum_{j=0}^{\rho}s_j(x)z^{\rho-j}y^j\}$, and its Gröbner basis $Q_0(x,y,z),\ldots,Q_\rho(x,y,z)$ such that any $Q(x,y,z)\in M_{\rho,r}$ can be represented as $S(x,y,z)=\sum_{j=0}^{\rho}Q_j(x,y,z)p_j(x)$. One of polynomials $Q_j(x,y,z)$ is guaranteed to satisfy the weighted degree constraint.

The required Gröbner basis can be found by the iterative interpolation algorithm [2], if one replaces its initialization stage with $Q_j(x,y,z):=z^{\rho-j}y^j$. This requires $O(n^2r^5)$ operations. Since (7) allows using much smaller r compared to the case of Guruswami-Sudan algorithm, substantial complexity reduction can be achieved. However, the complexity still remains quite high for a practical implementation.

We propose to extend the binary interpolation algorithm proposed in [5] to the case of partially homogenized polynomials. The main idea of the proposed method is to begin with a module of low-degree polynomials having roots of small multiplicity, and use them to obtain a module of polynomials of higher degree with roots of larger multiplicity. The following lemma gives the starting point for this sequence of modules.

Lemma 4. *Let $q_{11}(x)$ and $q_{10}(x)$ be coprime polynomials. Then $M_{1,1}=\widehat{M}$, where $\widehat{M}=[\phi(x)z,\phi(x)y,q_{11}(x)z+q_{10}(x)y]$*

Proof: Let us first construct linearly independent (over $\mathbb{F}[x]$) polynomials generating module $\widehat{M}$, and then show that they indeed generate $M_{1,1}$.

The extended Euclidean algorithm can be used to derive the polynomials $u_{00}(x),u_{10}(x),u_{01}(x),u_{11}(x)$, such that

$$g_{11}(x)=\gcd(\phi(x),q_{10}(x))=u_{10}(x)\phi(x)+u_{11}(x)q_{10}(x), \tag{8}$$

and

$$0=u_{00}(x)\phi(x)+u_{01}(x)q_{10}(x). \tag{9}$$

Let

$$\begin{aligned}\tilde{G}_0(x,y,z)&=u_{00}(x)\phi(x)y+u_{01}(x)(q_{11}(x)z+q_{10}(x)y)\\&=u_{01}(x)q_{11}(x)z,\\G_1(x,y,z)&=u_{10}(x)\phi(x)y+u_{11}(x)(q_{11}(x)z+q_{10}(x)y)\\&=u_{11}(x)q_{11}(x)z+g_{11}(x)y.\end{aligned}$$

These polynomials together with $\phi(x)z$ represent another basis of $\widehat{M}$. Let us further introduce the polynomial $G_0(x,y,z)=\gcd(\phi(x),u_{01}(x)q_{11}(x))z$. It can be seen that $u_{01}(x)q_{11}(x)=\frac{\phi(x)u_{00}(x)q_{11}(x)}{q_{10}(x)}=\frac{\phi(x)}{g_{11}(x)}\frac{q_{11}(x)u_{00}(x)}{q'_{10}(x)}$, where $q_{10}(x)=g_{11}(x)q'_{10}(x)$. Since $(q'_{10}(x),\phi(x))=1$, from (9) one obtains that $q'_{10}(x)|u_{00}(x)$, i.e. $u_{00}(x)=q'_{10}(x)u'_{00}(x)$. Then (9) implies $u'_{00}(x)\phi(x)=u_{01}(x)g_{11}(x)$. It follows from the properties of the extended Euclidean algorithm that $u_{00}(x)u_{11}(x)-u_{10}(x)u_{01}(x)=(-1)^i$ for some i. Hence,

$$u'_{00}(x)\left(q'_{10}(x)u_{11}(x)-u_{10}(x)\frac{\phi(x)}{g_{11}(x)}\right)=(-1)^i.$$

Therefore, $u'_{00}(x)=\pm1$ and $u_{01}(x)q_{11}(x)=\pm\frac{\phi(x)}{g_{11}(x)}q_{11}(x)$. Hence,

$$G_0(x,y,z)=\frac{\phi(x)}{g_{11}(x)}(g_{11}(x),q_{11}(x))z=\frac{\phi(x)}{g_{11}(x)}z.$$

The last equality is due to coprimeness of $q_{11}(x)$ and $q_{00}(x)$. Since the transformations used to obtain $G_0(x,y,z)$ and $G_1(x,y,z)$ from $\phi(x)z$, $\phi(x)y$, and $q_{11}(x)z-q_{10}(x)y$ are invertible, they generate the same module $\widehat{M}$.

Let $A(x,y,z)=u(x)z+v(x)y$ be a polynomial in $M_{1,1}$, i.e. $u(x)q_{10}(x)-v(x)q_{11}(x)=a(x)\phi(x)$ for some $a(x)$. Since

$g_{11}(x)|q_{10}(x)$, $g_{11}(x)|\phi(x)$ and $\gcd(q_{11}(x), q_{10}(x)) = 1$, $v(x)$ is divisible by $g_{11}(x)$. Consider the polynomial

$$\begin{aligned}
R(x,y,z) &= A(x,y,z) - \frac{v(x)}{g_{11}(x)} G_1(x,y,z) \\
&= z\left(u(x) - \frac{v(x)u_{11}(x)q_{11}(x)}{g_{11}(x)}\right) \\
&= z\left(u(x) - \frac{u_{11}(x)}{g_{11}(x)}(u(x)q_{10}(x) - a(x)\phi(x))\right) \\
&= z\left(u(x)\left(1 + \frac{u_{10}(x)\phi(x) - g_{11}(x)}{g_{11}(x)}\right)\right. \\
&\quad \left. + \frac{a(x)u_{11}(x)\phi(x)}{g_{11}(x)}\right) \\
&= z\frac{\phi(x)}{g_{11}(x)}\left(u_{10}(x)u(x) + a(x)u_{11}(x)\right).
\end{aligned}$$

This polynomial is divisible by $G_0(x,y,z)$. Hence, an arbitrary polynomial $A(x,y,z) \in M_{1,1}$ can be expressed via $G_0(x,y,z), G_1(x,y,z)$, the basis polynomials of $\widehat{M}$. Therefore $M_{1,1} \subset \widehat{M}$. The inclusion $\widehat{M} \subset M_{1,1}$ follows from the definition of $M_{1,1} \subset I_1$. ■

The required interpolation polynomial can be found in $M_{\rho,r}$, where the parameters ρ and r must satisfy the constraints derived in section IV. The proposed approach consists in construction of a sequence of modules $M_{u,v}$, which are obtained via two operations:

- module expansion, which obtains $M_{u+1,v}$ from $M_{u,v}$;
- root accumulation, which obtains $M_{u'+u'',v'+v''}$ from $M_{u',v'}$ and $M_{u'',v''}$.

These operations are described in the following subsections.

B. Module expansion

The following lemma reveals a useful property of Gröbner bases of $M_{\rho,r}$ with respect to lexicographic ($y \prec z \prec x$) monomial ordering.

Lemma 5. *Let $Q_0(x,y,z), \ldots, Q_\rho(x,y,z)$ be a Gröbner basis of $M_{\rho,r}$ with respect to lexicographic term ordering, where $\rho \geq r$. Then $Q_\rho(x,y,z) = g_{11}^r(x)y^\rho + Q'(x,y,z)$, where $g_{11}(x)$ is given by* (8)*, and $Q'(x,y,z)$ is not divisible by y^ρ.*

Proof: The polynomials in the considered basis are given by $Q_i(x,y,z) = \sum_{j=0}^{i} q_{ji}(x)y^j z^{\rho-j}$. $Q_\rho(x,y,z) = \sum_{j=0}^{\rho} q_{j,\rho}(x)y^j z^{\rho-j}$ is the only polynomial in the considered basis having terms divisible by y^ρ. It has roots $(x_i, \alpha q_{11}(x_i), -\alpha q_{10}(x_i))$ of multiplicity r. For all i such that $q_{10}(x_i) = g_{11}(x_i) = 0$ this implies that the polynomial $\tilde{Q}_\rho(x,\theta) = \sum_{j=0}^{\rho} q_{\rho-j,\rho}(x)\theta^j$ has roots $(x_i, 0)$ of multiplicity r. Hence, $(x-x_i)^r | q_{\rho,\rho}(x)$. Since $g_{11} = \prod_{i:q_{1,1}(x_i)=0}(x-x_i)$, one obtains $g_{11}^r | q_{\rho,\rho}(x)$. On the other hand, $z^{\rho-r}G_1^r(x,y,z) \in M_{\rho,r}$, where $G_1(x,y,z) = u_{11}(x)q_{11}(x)z + g_{11}(x)y$. Since $Q_0(x,y,z), \ldots, Q_\rho(x,y,z)$ is a Gröbner basis with respect to lexicographic term ordering $q_{\rho,\rho}(x)|g_{11}^r(x)$. Hence, $q_{\rho,\rho}(x) = g_{11}^r(x)$. ■

This result is similar to the well-known property of zero-dimensional ideals of $\mathbb{F}[x,y]$, which must contain polynomials $Q'(x,y), Q''(x,y)$, such that $\operatorname{LT} Q'(x,y) = y^a, \operatorname{LT} Q''(x,y) = x^b$ in any Gröbner basis.

The following lemma provides a simple property, which can be used to check if one has obtained a Gröbner basis of the required module.

Lemma 6. *Let $Q_j(x,y,z), j = 0..\rho$ be polynomials such that $Q_j(x_i, \alpha q_{11}(x_i), \alpha q_{01}(x_i)) = 0^r$, and $\operatorname{ydeg} Q_j(x,y,z) = j, j = 0..\rho$. If $\Delta((Q_0(x,y,z), \ldots, Q_\rho(x,y,z)) = n\frac{r(r+1)}{2}$, then these polynomials constitute a Gröbner basis of $M_{\rho,r}$.*

Proof: The proof is similar to the one of Lemma 6 in [5]. ■

The main difference of the Wu algorithm compared to the Guruswami-Sudan one is that one should care not only about root multiplicity r, but also about polynomial degree ρ. The interpolation algorithm presented below involves two types of operations: increasing both r and ρ, and increasing ρ only. The implementation of the latter operation is based on the following lemma.

Lemma 7. *Consider the module $M_{\rho,r} = [Q_0(x,y,z), \ldots, Q_\rho(x,y,z)]$. Then $M_{\rho+1,r} = [zQ_0(x,y,z), \ldots, zQ_\rho(x,y,z), yQ_0(x,y,z), \ldots, yQ_\rho(x,y,z)]$*

Proof: Assume without loss of generality that $Q_0(x,y,z), \ldots, Q_\rho(x,y,z)$ is a Gröbner basis of $M_{\rho,r}$ with respect to lexicographic ordering. The polynomials $yQ_0(x,y,z), \ldots, yQ_\rho(x,y,z)$ generate some submodule of $M_{\rho+1,r}$. Any polynomial $A(x,y,z) \in M_{\rho+1,r}$ can be represented as $A(x,y,z) = a_{\rho+1}(x)y^{\rho+1} + zA'(x,y,z)$, where $A'(x,y,z)$ does not contain terms divisible by $y^{\rho+1}$. By lemma 5, $g_{11}^r(x)|a_{\rho+1}(x)$, and $yQ_\rho(x,y) = g_{11}^r(x)y^{\rho+1} + \ldots$. Therefore, dividing $A(x,y,z)$ by $yQ_0(x,y,z), \ldots, yQ_\rho(x,y,z)$ one obtains a remainder $zR(x,y,z)$, where $R(x,y,z) \in M_{\rho,r}$. Hence, there exist $q_0(x), \ldots, q_\rho(x) : zR(x,y,z) = \sum_{j=0}^{\rho} zQ_j(x,y,z)q_j(x)$. ■

Observe that the basis given in the statement of the above lemma is highly redundant, since at most $\rho+2$ elements of $M_{\rho+1,r}$ can be linearly independent over $\mathbb{F}[x]$. This problem can be avoided by employing the randomized module transformation method originally proposed in [5] in the context of fast ideal multiplication. Namely, one can construct a sequence of modules $M_{\rho+1,r}^{(j)} = \{S(x,y,z) = P(x,y,z) + a(x)P_j(x,y,z)|a(x) \in \mathbb{F}[x], P(x,y,z) \in M_{\rho+1,r}^{(j-1)}\}$, where $M_{\rho+1,r}^{(0)} = [zQ_0(x,y,z), \ldots, zQ_\rho(x,y,z), yQ_\rho(x,y,z)]$, the polynomials $P_j(x,y,z)$ are computed as $P_j(x,y,z) = y\sum_{i=0}^{\rho} \beta_{ij}Q_j(x,y,z)$, where β_{ij} are independent random values uniformly distributed over $\mathbb{F}$, and $Q_j(x,y,z)$ are the basis elements of $M_{\rho,r}$. This process can be terminated as soon as the condition of Lemma 6 is satisfied. Figure 3 illustrates the proposed algorithm. The described approach requires computing at each step a Gröbner basis $(S_0(x,y,z), \ldots, S_{\rho+1}(x,y,z))$ of $M_{\rho+1,r}^{(j)}$. This can be implemented with the multidimensional Euclidean algorithm shown in Figure 2. It assumes that $\operatorname{ydeg} S_j(x,y,z) = j, 0 \leq j < i$, and produces a Gröbner basis with the same property.

Theorem 2. *Given a Gröbner basis of $M_{\rho,r}$, such that $\operatorname{LT} S_i(x,y,z) = z^{\rho-i}y^i x^{u_i}$, algorithm Expand constructs a*

```
Reduce$((S_0(x,y),\dots,S_{i-1}(x,y)),P(x,y))$
 1  $S_i(x,y) \leftarrow P(x,y)$
 2  while $\exists j : (0 \le j < i) \wedge (\operatorname{ydeg} S_j(x,y) = \operatorname{ydeg} S_i(x,y))$
 3  do if $\operatorname{LT} S_i(x,y) | \operatorname{LT} S_j(x,y)$
 4        then $W(x,y) \leftarrow S_j(x,y) - \frac{\operatorname{LT} S_j(x,y)}{\operatorname{LT} S_i(x,y)} S_i(x,y)$
 5             $S_j(x,y) \leftarrow S_i(x,y)$
 6             $S_i(x,y) \leftarrow W(x,y)$
 7        else $S_i(x,y) \leftarrow S_i(x,y) - \frac{\operatorname{LT} S_i(x,y)}{\operatorname{LT} S_j(x,y)} S_j(x,y)$
 8  if $S_i(x,y) = 0$
 9     then $i \leftarrow i - 1$
10  return $(S_0(x,y),\dots,S_i(x,y))$
```

Fig. 2. Construction of a Gröbner basis of $\mathcal{M}' = \{S(x,y,z) + a(x)P(x,y,z) | S(x,y,z) \in \mathcal{M}\}$ out of a Gröbner basis $(S_0(x,y,z),\dots,S_{i-1}(x,y,z))$ of $\mathcal{M}$

```
Expand$((S_0(x,y,z),\dots,S_\rho(x,y,z)),n,r)$
1  $\mathcal{G}_0 \leftarrow (zS_0(x,y,z),\dots,zS_\rho(x,y,z),yS_\rho(x,y,z))$
2  $j \leftarrow 1$
3  while $\Delta(\mathcal{G}) > n\frac{r(r+1)}{2}$
4  do $\beta_{ij} \leftarrow \mathrm{RAND}(), 0 \le i \le \rho$
5     $P_j(x,y,z) \leftarrow y\sum_{i=0}^{\rho} \beta_{ij} \cdot S_i(x,y,z)$
6     $\mathcal{G}_j \leftarrow$ Reduce$(\mathcal{G}_{j-1}, P_j(x,y,z))$
7     $j \leftarrow j + 1$
8  return $\mathcal{G}$
```

Fig. 3. Construction of a Gröbner basis of $M_{\rho+1,r}$ out of a Gröbner basis of $M_{\rho,r}$

Gröbner basis of $M_{\rho+1,r}$ with average complexity $O(n^2r^4/\rho)$.

Proof: The proof essentially follows the analysis in [5]. The initial basis $\mathcal{G}_0$ constructed on line 1 satisfies the input constraints of the multidimensional Euclidean algorithm, so on each iteration it indeed produces a Gröbner basis of $M_{\rho+1,r}^{(j)}$. If sufficiently many iterations are performed, then the linear transformation given by random values β_{ij} is invertible, so that one can reconstruct all polynomials $yS_i(x,y,z)$ from the polynomials $P_j(x,y,z)$. Hence, the algorithm converges eventually to a Gröbner basis of $M_{\rho+1,r}$.

To estimate the convergence rate, let us compute the probability that $M_{\rho+1,r}$ is generated by $zS_{i'}(x,y,z), yS_\rho(x,y,z), P_{i''}(x,y,z)$, $0 \le i' \le \rho$, $1 \le i'' \le \delta$. These polynomials can be represented as $(z^{\rho+1}, z^\rho y, \dots, y^{\rho+1})\mathbf{G}(x)$, where $\mathbf{G}(x)$ is a $(\rho+1)\times(\rho+1+\delta)$ polynomial matrix. The true Gröbner basis can be represented in a similar way as a $(\rho+1)\times(\rho+1)$ matrix $\mathbf{B}(x)$, which satisfies $\mathbf{G}(x)\mathbf{U}(x) = \mathbf{B}(x)$ for some polynomial matrix $\mathbf{U}(x)$. On the other hand, the above polynomials are in $M_{\rho+1,r}$, so $\mathbf{G}(x) = \mathbf{B}(x)\mathbf{W}(x)$. This implies that $\mathbf{W}(x)\mathbf{U}(x) = I$, i.e. for any x_0 in $\bar{\mathbb{F}}$, the algebraic closure of $\mathbb{F}$, $\mathbf{W}(x_0)$ must be a full-rank matrix. Consider a decomposition $\mathbf{W}(x) = (\tilde{\mathbf{W}}(x)|\widehat{\mathbf{W}}(x))$, where $\tilde{\mathbf{W}}(x)$ is the $(\rho+2)\times(\rho+2)$ submatrix given by first $\rho+2$ columns of $\mathbf{W}(x)$. One can identify at most $N = \deg\det\tilde{\mathbf{W}}(x)$ pairs of eigenvalues $x_i \in \mathbb{F}^{\sigma_i}$, where $\mathbb{F}^{\sigma_i}$ are some algebraic extensions of $\mathbb{F}$, and corresponding linearly independent eigenvectors $v_i \in (\mathbb{F}^{\sigma_i})^{\rho+1}$, such that $v_i\mathbf{W}(x_i) = 0$. Assuming the entries of $W(x_i)$ to be independent random variables uniformly distributed over $\mathbb{F}^{\sigma_i}$, one obtains (see [5] for details) the following expression for probability of $\mathbf{W}(x)$ being a full-rank matrix for all x:

$$\Theta(\delta) = \sum_{\omega} P_\omega \prod_{j=1}^{N} \left(1 - \frac{1}{|\,\mathbb{F}\,|^{j\delta}}\right)^{j\omega_j},$$

where summation is performed over all partitions ω such that $\sum_j j\omega_j = N$, and P_ω is the probability of obtaining a factorization of $\det\tilde{\mathbf{W}}(x)$ into irreducible over $\mathbb{F}$ polynomials $\phi_s(x)$, such that exactly ω_j of them have degree j, i.e. their roots x_i are in $\mathbb{F}^j$. This expression is dominated by the multiples corresponding to $j = 1$, so the probability of $M_{\rho+1,r}$ not being generated by $\rho+1+\delta$ polynomials decreases exponentially fast with δ, i.e. *Expand* converges in $O(1)$ iterations.

Observe that $\det\tilde{\mathbf{W}}(x) = \frac{\det\tilde{\mathbf{G}}(x)}{\det\mathbf{B}(x)}$, i.e. $N = \Delta(\mathcal{G}_0) - n\frac{r(r+1)}{2}$, where $\tilde{\mathbf{G}}(x)$ corresponds to the initial approximation $\mathcal{G}_0$ constructed on line 1. Assuming that all polynomials $S_i(x,y,z)$ in the original Gröbner basis of $M_{\rho,r}$ have the same $(1,w_1,w_2)$-weighted degree C, one obtains that their leading terms satisfy $\operatorname{LT} S_i(x,y,z) = z^{\rho-i}y^ix^{u_i}$, so that $u_i + iw_1 + (\rho-i)w_2 = C$ and $\sum_{i=0}^{\rho} u_i = n\frac{r(r+1)}{2}$. This implies $C = (w_1+w_2)\frac{\rho}{2} + n\frac{r(r+1)}{2(\rho+1)}$ and $u_i = (w_1-w_2)\frac{\rho}{2} + n\frac{r(r+1)}{2(\rho+1)} + i(w_2-w_1)$. Hence, $\Delta(\mathcal{G}_0) = \sum_{i=0}^{\rho} u_i + u_\rho = n\frac{r(r+1)}{2} + n\frac{r(r+1)}{2(\rho+1)} + \frac{\rho}{2}(w_2-w_1)$, i.e. $N = O(nr^2/\rho)$. During the *WHILE* loop, the weighted degree of polynomials in $\mathcal{G}_j$ decreases from C to $C' = (w_1+w_2)\frac{\rho+1}{2} + n\frac{r(r+1)}{2(\rho+2)}$. Each polynomial contains at most $\rho+2$ monomials of any fixed $(1,w_1,w_2)$-weighted degree between C and C'. Hence, the total number of monomials to be cancelled is $n\frac{r(r+1)}{2(\rho+1)} - \frac{w_1+w_2}{2}(\rho+2)$. Assuming that each iteration of *Reduce* eliminates exactly one of them by summing together appropriate polynomials, which contain $O(nr^2)$ terms, the total complexity of *Expand* can be estimated as $O(n^2r^4/\rho)$. ■

C. Root accumulation

The proposed root accumulation method essentially follows the one proposed in [5] for the case of Guruswami-Sudan algorithm.

Lemma 8. *Let $M_{\rho_1,r_1} = [S_0(x,y,z),\dots,S_{\rho_1}(x,y,z)]$ and $M_{\rho_2,r_2} = [P_0(x,y,z),\dots,P_{\rho_2}(x,y,z)]$ be the modules given by their Gröbner bases satisfying the constraints of Lemma 6. Then*

$$M_{\rho_1+\rho_2,r_1+r_2} = [S_i(x,y,z)P_j(x,y,z), i = 0..\rho_1, j = 0..\rho_2]. \tag{10}$$

Proof: See [5, Lemma 7]. ■

This lemma allows one to generalize the fast ideal multiplication algorithm proposed in [5] to the case of the rational curve fitting problem. Namely, one can replace pairwise poly-

```
MERGE($(S_i(x,y,z), i=0..\rho_1), (P_i(x,y,z), i=0..\rho_2), \Delta_0$)
1  for $i \leftarrow 0$ to $\rho_1+\rho_2$
2  do $Q_i(x,y,z) = \min_{0\leq j\leq v} P_{i-j}(x,y,z)S_j(x,y,z)$
3  $\mathcal{B} = (Q_0(x,y,z),\ldots,Q_{\rho_1+\rho_2}(x,y,z))$
4  while $\Delta(\mathcal{B}) > \Delta_0$
5  do $\alpha_i \leftarrow rand(), 0\leq i\leq \rho_1$
6      $\beta_j \leftarrow rand(), 0\leq j\leq \rho_2$
7      $Q(x,y,z) \leftarrow (\sum_{i=0}^{\rho_1}\alpha_i S_i(x,y,z))(\sum_{i=0}^{\rho_2}\beta_i P_i(x,y,z))$
8      $\mathcal{B} \leftarrow$ REDUCE($\mathcal{B}, Q(x,y,z)$)
9  return $\mathcal{B}$
```

Fig. 4. Construction of a Gröbner basis of $M_{\rho_1+\rho_2,r_1+r_2}$.

nomial products in (10) with sufficiently many polynomials

$$Q_j(x,y,z) = \left(\sum_{i=0}^{\rho_1}\alpha_{ij}S_i(x,y,z)\right)\left(\sum_{i=0}^{\rho_2}\beta_{ij}P_i(x,y,z)\right),$$

where α_{ij},β_{ij} are random values uniformly distributed over $\mathbb{F}$. Then the sequence $\mathcal{M}^{(j+1)} = \{Q(x,y,z) + a(x)Q_j(x,y,z)|Q(x,y,z) \in \mathcal{M}^{(j)}\}$, where $\mathcal{M}^{(0)} \subset M_{\rho_1+\rho_2,r_1+r_2}$, converges to $M_{\rho_1+\rho_2,r_1+r_2}$. It is reasonable to construct the initial submodule $\mathcal{M}^{(0)}$ in some simple way. For example, it can be defined as a module generated by polynomials $S_{i-j_i}(x,y,z)P_{j_i}(x,y,z), i=0..\rho_1+\rho_2$, where j_i are selected so that the leading term of the obtained product is minimized.

Figure 4 presents the algorithm implementing this approach. One should set $\Delta_0 = n\frac{r(r+1)}{2}, r = r_1+r_2$, so that the $WHILE$ loop terminates as soon as $\Delta(\mathcal{B}) = \Delta_0$. This condition indicates that the module $\mathcal{M}^{(j)}$, generated by the recently obtained Gröbner basis $\mathcal{B}$, is equal to $M_{\rho_1+\rho_2,r_1+r_2}$. Using the techniques described in [5], one obtains that in the case of $r_1=r_2=r, \rho_1=\rho_2=\rho$ the algorithm converges in average in $O(1)$ iterations. The number of operations needed to multiply two polynomials is given by $O(\rho nr\log\rho\log(nr))$, and the average number of iterations performed by $Reduce$ is equal $nr\frac{\rho-r}{\rho+1}$, so the average complexity is given by $O(n^2r^3\frac{\rho-r}{\rho+1})$.

In the case of $\mathbb{F}$ being a field of characteristic 2 and $S_i(x,y,z) = P_i(x,y,z), 0\leq i\leq \rho_1=\rho_2$, the complexity of the above algorithm can be slightly reduced by constructing $Q_{2j}(x,y,z) = S_i^2(x,y,z)$, and using the original expression given on line 2 of the algorithm only for odd i.

D. Binary interpolation algorithm

The algorithms presented above can be used to construct a Gröbner basis of $M_{\rho,r}$ from the one of $M_{1,1}$ via a sequence of intermediate modules M_{ρ_i,r_i}. Theorem 2 suggests that ρ_i should be kept as high as possible in order to minimize the overall complexity[2]. That is, one should keep the ratio ρ_i/r_i as close as possible to the final value ρ/r, while respecting the integrality constraints on r_i and ρ_i. Figure 5 presents the algorithm implementing this approach. $(1,w_1,w_2)$-weighted degree lexicographic ordering with $y\prec z\prec c$ should be used throughout this algorithm. The algorithm starts by construction of a Gröbner basis of $M_{1,1}$ (lines 1–2) using the result of lemma 4. $Reduce$ algorithm is used to obtain two linearly independent over $\mathbb{F}[x]$ polynomials being a Gröbner basis of this module. Then this module is lifted to $M_{\lfloor\rho/r\rfloor,1}$, which is then used as input to the binary exponentiation algorithm. The variables R and Π correspond to r_i and ρ_i, respectively.

```
INTERPOLATE($q_{10}(x), q_{11}(x), \phi(x), n, r, \rho$)
 1  $\mathcal{G} \leftarrow (z\phi(x), y\phi(x))$
 2  $\mathcal{G} \leftarrow$ REDUCE($\mathcal{G}, zq_{11}(x) - yq_{10}(x)$)
 3  $\pi \leftarrow \lfloor\frac{\rho}{r}\rfloor$
 4  for $j \leftarrow 1$ to $\pi$
 5  do $\tilde{\mathcal{G}} =$ EXPAND($\mathcal{G}, n, 1$)
 6  $\Pi = \pi$
 7  $\mathcal{B} \leftarrow \mathcal{G}$
 8   Let $r = \sum_{j=0}^{m} r_j2^j, r_j \in \{0,1\}$
 9  $R \leftarrow 1$
10  for $j \leftarrow m-1$ to $0$
11  do $R \leftarrow 2R$
12      $\Pi = 2\Pi$
13      $\mathcal{B} \leftarrow$ MERGE($\mathcal{B},\mathcal{B}, nR(R+1)/2$)
14      if $r_j = 1$
15        then $R \leftarrow R+1$
16             $\Pi \leftarrow \Pi + \pi$
17             $\mathcal{B} \leftarrow$ MERGE($\mathcal{B},\mathcal{G}, nR(R+1)/2$)
18      while $\lfloor R\rho/r\rfloor > \Pi$
19      do $\mathcal{B} \leftarrow$ EXPAND($\mathcal{B}, n, R$)
20         $\Pi \leftarrow \Pi + 1$
21  return $\mathcal{B}$
```

Fig. 5. Construction of a Gröbner basis for $M_{\rho,r}$

The complexity of this algorithm is dominated by the last iteration, so it can be estimated as $O(n^2r^3)$. This comes mostly from the multi-dimensional Euclidean algorithm ($Reduce$). It is possible to speed up the proposed algorithm by employing the generalization of Knuth-Schönhage algorithm given in [13].

The input to the proposed algorithm should be either the polynomials $q_{10}(x), q_{11}(x)$ derived in Section III, or the polynomials $\Lambda(x), xB(x)$ obtained from the Berlekamp-Massey algorithm, as in the original Wu method. Given the output of this algorithm, one should select the smallest polynomial $S(x,y,z)$ in the obtained Gröbner basis with respect to $(1,w_1,w_2)$-weighted degree lexicographic term ordering, and recover all polynomials $a(x), b(x)$, such that $S(x,a(x),b(x)) = 0$. This can be implemented using the generalization of the Roth-Ruckenstein algorithm presented in [6].

E. List decoding with side information

Wu list decoding algorithm was shown in [14] to be able to take into account information about error-free positions. Namely, one may be interested in finding $\sigma(x)$ such that its roots are in some subset Θ of the original set of code locators $\{x_1,\ldots,x_n\}$. This enables one to obtain higher error

[2] This was not implemented in the initial version of the algorithm presented in [12]. As a result, the complexity reduction provided by the algorithm given in this paper is much more significant.

correction capability. The proposed interpolation algorithm extends naturally to this case by replacing $\phi(x)$ on line 1 with $\phi'(x) = \prod_{x_j \in \Theta}(x - x_j)$.

VI. Numeric results

The re-formulated Wu decoding method together with the above described binary interpolation algorithm have been implemented in C++ programming language, and computer simulations[3] were used to investigate their complexity. For the sake of comparison, the iterative interpolation algorithm [2] and Guruswami-Sudan decoding method with binary interpolation and re-encoding [5] were also implemented. Observe that the latter algorithm requires different root multiplicity. In all cases root multiplicity r was set to the smallest value allowing correction of t errors. The obtained results are given in Table I.

It can be seen that in all cases the implementation of Wu decoding method based on the proposed binary interpolation algorithm outperforms the one based on IIA at least by a factor of five. Furthermore, since Wu method requires much smaller root multiplicity r than in the case of Guruswami-Sudan method, it outperforms even its most efficient implementation, which is based on the binary interpolation algorithm and re-encoding trick [4]. However, in some cases the implementation of the Wu decoder based on IIA turns out to be slower compared to the Guruswami-Sudan algorithm with re-encoding utilizing the binary interpolation algorithm.

VII. Conclusions

In this paper an efficient interpolation algorithm for the Wu list decoding method [6] was given. The interpolation step was formulated as construction of a partially homogenized trivariate polynomial. This avoids the problem of roots at infinity, which arises in the original description of the method, and enables application of the fast interpolation algorithm based on the binary exponentiation method. Furthermore, improved estimates for the parameters of the Wu method were derived. These estimates, as well as the proposed interpolation algorithm, which is an extension of the one given in [5], can be applied to the original Wu method based on the Berlekamp-Massey algorithm as well.

Numeric results indicate that the proposed approach enables complexity reduction by a factor at least five compared to the implementation based on the iterative interpolation algorithm. In all cases the Wu list decoding method based on the binary interpolation algorithm outperforms the most efficient existing implementation of the Guruswami-Sudan algorithm.

Acknowledgment

The authors thank the anonymous reviewers for many helpful comments, which have greatly improved the quality of the paper.

This work was supported in part by Russian Ministry of Education and Science under the contract 07.514.11.4101, Korean World Class University (WCU) project WCUR32-2012-000-20014-0, and Korean National Research Foundation under the projects FR2010-0020942, MEST2012-002521.

References

[1] V. Guruswami and M. Sudan, "Improved decoding of Reed-Solomon and algebraic-geometric codes," *IEEE Transactions on Information Theory*, vol. 45, no. 6, pp. 1757–1767, September 1999.

[2] R. R. Nielsen and T. Hoholdt, "Decoding Reed-Solomon codes beyond half the minimum distance," in *Proceedings of the International Conference on Coding Theory and Cryptography*. Mexico: Springer-Verlag, 1998, pp. 221–236.

[3] R. Roth and G. Ruckenstein, "Efficient decoding of Reed-Solomon codes beyond half the minimum distance," *IEEE Transactions on Information Theory*, vol. 46, no. 1, pp. 246–257, 2000.

[4] R. Koetter and A. Vardy, "A complexity reducing transformation in algebraic list decoding of Reed-Solomon codes," in *Proceedings of ITW2003*, March 2003, pp. 10–13.

[5] P. Trifonov, "Efficient interpolation in the Guruswami-Sudan algorithm," *IEEE Transactions on Information Theory*, vol. 56, no. 9, pp. 4341–4349, September 2010.

[6] Y. Wu, "New list decoding algorithms for Reed-Solomon and BCH codes," *IEEE Transactions On Information Theory*, vol. 54, no. 8, August 2008.

[7] T. Becker and V. Weispfenning, *Gröbner Bases. A Computational Approach to Commutative Algebra*. New York: Springer, 1993.

[8] S. Gao, "A new algorithm for decoding Reed-Solomon codes," in *Communications, Information and Network Security*, V. Bhargava, H. V. Poor, V. Tarokh, and S. Yoon, Eds. Kluwer, 2003, pp. 55–68.

[9] M. Ali and M. Kuijper, "Minimal list decoding of Reed-Solomon codes using a parameterization of Gröbner bases," in *Proceedings of IEEE International Symposium on Information Theory*, 2011.

[10] K. Lee and M. E. O'Sullivan, "List decoding of Reed-Solomon codes from a Gröbner basis perspective," *Journal of Symbolic Computation*, vol. 43, no. 9, September 2008.

[11] T. Sauer, "Polynomial interpolation of minimal degree and Gröbner bases," in *Gröbner Bases and Applications (Proceedings of the International Conference "33 Years of Gröbner Bases")*, ser. London Mathematical Society Lecture Notes, B. Buchberger and F. Winkler, Eds., vol. 251. Cambridge University Press, 1998, pp. 483–494.

[12] P. Trifonov, "Another derivation of Wu list decoding algorithm and interpolation in rational curve fitting," in *Proceedings of IEEE R8 International Conference on Computational Technologies in Electrical and Electronics Engineering*, 2010, pp. 59–64.

[13] M. Alekhnovich, "Linear Diophantine equations over polynomials and soft decoding of Reed-Solomon codes," *IEEE Transactions On Information Theory*, vol. 51, no. 7, pp. 2257–2265, July 2005.

[14] Y. Wu, "Erasure-only list decoding of Reed-Solomon and BCH codes with applications to their product codes," in *Proceedings of IEEE International Symposium on Information Theory*, 2011.

PLACE
PHOTO
HERE

Peter Trifonov was born in St.Petersburg, USSR in 1980. He received the MSc degree in computer science in 2003, and PhD (Candidate of Science) degree from St.Petersburg State Polytechnic University in 2005. Currently he is an Associate Professor at the Distributed Computing and Networking department of the same university. His research interests include coding theory and its applications in telecommunications and other areas. Since January, 2012 he is serving as a vice-chair of the IEEE Russia Joint Sections Information Theory Society Chapter.

[3] Simulations were run on a computer based on Intel Core i7 920 CPU.

TABLE I
DECODING TIME, S

	$(255, 219), t = 19$	$(255, 128), t = 73$	$(31, 15), t = 10$	$(63, 31), t = 19$	$(63, 20), t = 28$
Wu+IIA [2]	0.56	1.1	0.33	0.073	7.88
Wu+binary	0.115	0.22	0.048	0.017	0.355
GS+re-encoding+binary [5]	1.94	0.83	0.11	0.065	–

PLACE PHOTO HERE

Moon Ho Lee Moon Ho Lee is a professor and former chair of the Department of Electronics Engineering in Chonbuk National University, Korea. He received the Ph.D. degree from Chonnam National University, Korea in 1984, and from the University of Tokyo, Japan in 1990, both Electrical Engineering, He was in University of Minnesota, U.S.A, from 1985 to 1986 as a post-doctor. He has been working in Namyang MBC broadcasting with chief engineer from 1970 to 1980, after then he joined to Chonbuk National University as a Professor. Dr. Lee has made significant original contributions in the areas of mobile communication code design, channel coding, and multi-dimensional source and channel coding. He has authored 34 books, 155 SCI papers in international journals, and 240 papers in domestic journals, and delivered 350 papers at international conferences. Dr. Lee is a member of the National Academy of Engineering in Korea and a Foreign Fellow of the Bulgaria Academy of Sciences. He is the inventor of Jacket Matrix and it in Wikipedia was cited over 49,559 times, Dec. 2011.