Another Derivation of Wu List Decoding Algorithm and Interpolation in Rational Curve Fitting

Peter Trifonov
Saint-Peterbsurg State Polytechnic University,
St.Petersburg, Russia,
petert@dcn.ftk.spbstu.ru

***Abstract*—A novel derivation of the Wu list decoding algorithm for Reed-Solomon codes is provided. The algorithm 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, 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 Q_i.
- $\operatorname{LT} Q$ is the leading term of the polynomial Q with respect to some term ordering.
- $\operatorname{ydeg} Q = j$ iff $\operatorname{LT} Q(x,y) = ax^u y^j$ (in the case of bivariate polynomials) or $\operatorname{LT} Q(x,y,z) = ax^u y^j z^{\rho-j}$ (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 $\operatorname{LT} Q(x,y,z) = ax^u y^j z^{\rho-j}$ for some $a \in \mathbb{F}$ and $u \in \mathbb{Z}$.
- $\Delta(\mathcal{B}) = \sum_{j=0}^{s} \operatorname{xdeg} \mathcal{B}_j$, where $\mathcal{B} = (B_0(x,y,z), \ldots, B_s(x,y,z))$ is a Gröbner basis of some module.

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 $\deg f(x) < k$, and $x_i \in \mathbb{F}$ are distinct code locators. Let $y_i = f(x_i) + e_i, i = 1..n$, be noisy symbols of the received word. The list decoding problem consists in finding all pairs $(f^{(j)}(x), \sigma^{(j)}(x))$, such that $\deg f^{(j)}(x) < k$ and $\sigma^{(j)}(x_i) = 0$ for at most t distinct x_i, so that $y_i\sigma^{(j)}(x_i) = f^{(j)}(x_i)\sigma^{(j)}(x_i), i = 1..n$. Here $f^{(j)}(x)$ identifies the corresponding codeword, and $\sigma^{(j)}(x)$ is the error polynomial. Observe that one can recover $\sigma^{(j)}(x)$ from $f^{(j)}(x)$ and vice versa.

Any bivariate polynomial $Q^{(j)}(x,y) = y\sigma^{(j)}(x) - f^{(j)}(x)\sigma^{(j)}(x)$ 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 $T(x) : T(x_i) = y_i$. Let the polynomials $Q'(x,y) = q_{00}(x) + yq_{10}(x)$ and $Q''(x,y) = q_{01}(x) + yq_{11}(x)$ be another basis of this module. Then

$$Q^{(j)}(x,y) = a^{(j)}(x)Q'(x,y) + b^{(j)}(x)Q''(x,y). \quad (1)$$

Since it is sufficient to find only the error polynomials corresponding to different solutions of the list decoding problem, one is interested in finding all pairs $(a^{(j)}(x), b^{(j)}(x))$, such that $\sigma^{(j)}(x) = a^{(j)}(x)q_{10}(x) + b^{(j)}(x)q_{11}(x)$ has at most t distinct roots x_i. One can consider only coprime polynomials, since any valid solution of the list decoding problem satisfies $Q^{(j)}(x, f^{(j)}(x)) = 0$, and if $c(x)$ divides both $a^{(j)}(x)$ and $b^{(j)}(x)$, then $Q^{(j)}(x,y) = c(x)\tilde{Q}^{(j)}(x,y)$, so that $\tilde{Q}^{(j)}(x, f^{(j)}(x)) = 0$.

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 $\operatorname{ydeg} Q''(x,y) = 1$. Then any valid $Q^{(j)}(x,y)$ is reducible to zero with respect to $Q'(x,y)$ and $Q''(x,y)$, and the polynomials $a^{(j)}(x)$ and $b^{(j)}(x)$ satisfying (1) can be recovered via the multivariate division algorithm. This implies that $\deg b^{(j)}(x) \le w_2 = t - \deg q_{11}(x)$, and $\deg a^{(j)}(x) \le w_1 = t + k - 1 - \deg q_{00}(x)$.

Hence, the problem of list decoding of Reed-Solomon code reduces to 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 early termination condition, as used in Gao decoding method [7].
3) [Rational curve fitting] Find all pairs of coprime polynomials $a^{(j)}(x), b^{(j)}(x) : \deg a^{(j)}(x) \le w_1 = t + k - 1 - \deg q_{00}(x), \deg b^{(j)}(x) \le w_2 = t - \deg q_{11}(x)$, such that
$$\sigma^{(j)}(x) = a^{(j)}(x)q_{10}(x) + b^{(j)}(x)q_{11}(x) \tag{2}$$
has at most t roots.
4) For each j reconstruct the codeword from symbols y_i such that x_i are not roots of $\sigma^{(j)}(x)$.

The described algorithm can be considered as a frequency-domain interpretation of the Wu method [6], which is based on the analytical continuation of the Berlekamp-Massey algorithm. Recall, that it consists in finding all pairs of polynomials $(\lambda(x), b(x))$, such that the error locator polynomial
$$\Lambda^*(x) = \lambda(x)\Lambda(x) + xB(x)b(x) \tag{3}$$
has at most t distinct roots, where $\Lambda(x)$ 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

It was suggested in [6] to solve the list decoding problem by finding a polynomial $Q(x,y)$ having roots $(x_i, -\frac{\Lambda(x_i^{-1})}{x_i^{-1}B(x_i^{-1})})$ of multiplicity r for some r. However, the existing bivariate interpolation algorithms cannot be immediately used to solve this problem, since most of them construct a basis of the ideal of polynomials with prescribed roots, and the described set is not an ideal if $B(x_i^{-1}) = 0$ for some i. Indeed, the polynomials $1-xy$ and $1-xy^2$ have a root $(0,\infty)$. However, the polynomial $(1-xy)-(1-xy^2) = xy^2 - xy$ does not have this root.

This difficulty can be avoided by introducing partially homogenized polynomials $S(x,y,z) = \sum_{j=0}^{\rho} s_j(x)z^{\rho-j}y^j$.

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

Proof: Assume without loss of generality that $z_0 \ne 0$. $S(x,y,z) = \sum_{j=0}^{\rho}\sum_{i\ge 0} s_{ji}x^iy^jz^{\rho-j}$ 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.
$$\sum_{i'\ge u}\sum_{j'=v}^{\rho-w}\binom{i'}{u}\binom{j'}{v}\binom{\rho-j'}{w}s_{j'i'}x_0^{i'-u}\frac{(\alpha y_0)^{j'-v}}{(\alpha z_0)^{j'+w-\rho}} = 0$$
for all $u,v,w \ge 0$, s.t. $u+v+w < r$. Then for $w = 0$ one obtains
$$z_0^{\rho-v}\sum_{i'\ge u}\sum_{j'=v}^{\rho}\binom{i'}{u}\binom{j'}{v}s_{j'i'}x_0^{i'-u}\,(y_0/z_0)^{j'-v} = 0, u+v < r$$
i.e. $(x_0, y_0/z_0)$ is a root of multiplicity r of $\hat{S}(x,\theta) = \sum_{j=0}^{\rho}\sum_{i\ge 0} s_{ji}x^i\theta^j$.

If $(x_0, y_0/z_0)$ is a root $\hat{S}(x,\theta)$ of multiplicity r, then $\hat{S}(x,\theta) = \sum_{u+v\ge r, v\le\rho} s^{[u,v]}(x-x_0)^u(\theta - y_0/z_0)^v$. Hence, $S(x,y,z) = z^{\rho}\hat{S}(x,y/z) = \displaystyle\sum_{u+v\ge r,v\le\rho}\frac{s^{[u,v]}}{z_0^v}(x-x_0)^u(yz_0 -$ $zy_0)^vz^{\rho-v} = \displaystyle\sum_{u+v\ge r,v\le\rho}\frac{s^{[u,v]}}{z_0^v}(x-x_0)^u((y-\alpha y_0)z_0 - (z -$ $\alpha z_0)y_0)^vz^{\rho-v}$. 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 Lemma 4 and Lemma 5 in [1].

Lemma 2. *Let $Q(x,y,z) = \sum_{j=0}^{\rho} q_j(x)y^jz^{\rho-j}$ 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 coprime polynomials such that $z_0a(x_0) + y_0b(x_0) = 0$, then $(x-x_0)^r | Q(x, a(x), b(x))$.*

Lemma 3. *Let $Q(x,y,z) = \sum_{j=0}^{\rho} q_j(x)y^jz^{\rho-j}$ be a polynomial such that $\operatorname{wdeg}_{(1,w_1,w_2)} S(x,y,z) < rt$, 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 $\deg a(x) \le w_1$, $\deg b(x) \le w_2$ and $z_ia(x_i) + y_ib(x_i) = 0$ for at least t points (x_i, y_i, z_i), then $S(x, a(x), b(x)) = 0$.*

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 $\rho > n\frac{r+1}{2t} - 1 = \frac{n}{t} - 1$. For $w > 0$ one obtains
$$\frac{2rt - w - \sqrt{D}}{2w} < \rho < \frac{2rt - w + \sqrt{D}}{2w}, \tag{4}$$

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

$$r > \frac{\left(n - t + \sqrt{n^2 - 2tn + wn}\right) w}{2(t^2 - wn)}. \qquad (5)$$

This can be satisfied if $t^2 - wn \geq 0$. Since for any Gröbner basis of $\mathcal{M}$ one has $\deg q_{11}(x) + \deg q_{00}(x) = n$ [5], one obtains $w = w_1 + w_2 = 2t + (k-1) - n$. Hence, decoding is possible if $t < n - \sqrt{n(k-1)}$. The bound for ρ given by (4) is much better than the one derived in [6] ($\rho = \lfloor \frac{rt}{w} \rfloor$), and applies to that algorithm as well. This immediately results in complexity reduction at all steps of both list decoding algorithms.

V. Efficient interpolation

As it was shown above, all pairs of polynomials $(a^{(j)}(x), b^{(j)}(x))$, such that the polynomial $\sigma^{(j)}(x) = a^{(j)}(x)q_{10}(x) + b^{(j)}(x)q_{11}(x)$ has t distinct roots, are given by the equation $S(x, a^{(j)}(x), b^{(j)}(x)) = 0$, 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. This polynomial must appear in a Gröbner basis of the ideal I_r of polynomials having these roots. However, the full Gröbner basis of this ideal contains a lot of polynomials not satisfying the constraint (4). 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 $O(n^2 r^5)$ operations. Since (5) 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 start from 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 $M_{1,1} = [\phi(x)z, \phi(x)y, q_{11}(x)z - q_{10}(x)y]$*

Proof: 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

$$g_{11}(x) = \gcd(\phi(x), q_{10}(x)) = u_{10}(x)\phi(x) - u_{11}(x)q_{10}(x), \qquad (6)$$

and

$$0 = u_{00}(x)\phi(x) - u_{01}(x)q_{10}(x). \qquad (7)$$

Let $\tilde{G}_0(x,y,z) = u_{00}(x)\phi(x)y + u_{01}(x)(q_{11}(x)z - q_{10}(x)y) = u_{01}(x)q_{11}(x)z$, $G_1(x,y,z) = u_{10}(x)\phi(x)y + u_{11}(x)(q_{11}(x)z - q_{10}(x)y) = u_{11}(x)q_{11}(x)z + g_{11}(x)y$. 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 $u_{01}(x)q_{11}(x) = \frac{\phi(x)q_{11}(x)u_{00}(x)}{q_{10}(x)} = \frac{\phi(x)}{g_{11}(x)} \frac{q_{11}(x)u_{00}(x)}{q'_{10}(x)}$, where $q_{10}(x) = g_{11}(x)q'_{10}(x)$. It follows from (7) that $q'_{10}(x)|u_{00}(x)$. The polynomials $\frac{q_{11}(x)u_{00}(x)}{q'_{10}(x)}$ and $g_{11}(x)$ are coprime. Hence, $G_0(x,y,z) = \frac{\phi(x)}{g_{11}(x)}z$. 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.

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 $g_{11}(x)$. Let $R(x,y,z) = A(x,y,z) + \frac{v(x)}{g_{11}(x)}G_1(x,y,z) = z\left(u(x) + \frac{v(x)u_{11}(x)q_{11}(x)}{g_{11}(x)}\right) = z\left(u(x) + \frac{u_{11}(x)}{g_{11}(x)}(u(x)q_{10}(x) - a(x)\phi(x))\right) = z\left(u(x)(1 + \frac{u_{10}(x)\phi(x) - g_{11}(x)}{g_{11}(x)}) - \frac{a(x)u_{11}(x)\phi(x)}{g_{11}(x)}\right) = z\frac{\phi(x)}{g_{11}(x)}\left(u_{10}(x)u(x) - a(x)u_{11}(x)\right)$. This polynomial is divisible by $G_0(x,y,z)$. ■

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* (6)*, and $Q'(x,y,z)$ is not divisible by y^ρ.*

Proof: $Q_\rho(x,y,z) = \sum_{j=0}^{\rho} q_{j,\rho}(x) y^j z^{\rho-j}$ 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 $i : q_{10}(x_i) = g_{11}(x_i) = 0$ 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, $z^{\rho-r}G_1^r(x,y,z) \in M_{\rho,r}$, i.e. $q_{\rho,\rho}(x) | g_{11}^r(x)$. Hence, $q_{\rho,\rho}(x) = g_{11}^r(x)$. ■

The next 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]. ■

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

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

Fig. 1. Construction of a Gröbner basis of $M_{\rho_1+\rho_2,r_1+r_2}$.

of $M_{\rho+1,r}$. Any polynomial $A(x,y,z) \in M_{\rho+1,r}$ can be represented as $A(x,y,z) = a_r(x)y^{\rho+1} + zA'(x,y,z)$, where $A'(x,y,z)$ is not divisible by $y^{\rho+1}$. By lemma 5, $g_{11}^r(x)|a_r(x)$. 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 $\mathbb{F}[x]$. Hence, we propose to 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 constructed 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 $M_{\rho,r}$. For each j one can construct a Gröbner basis of $M_{\rho+1,r}^{(j)}$ using the *Reduce* algorithm given in [5]. Obviously, one can recover $yQ_0(x,y,z),\ldots,yQ_\rho(x,y,z)$ if sufficiently many polynomials $P_j(x,y,z)$ are constructed. Hence, the sequence $M_{\rho+1,r}^{(j)}$ converges eventually to $M_{\rho+1,r}$. In practice, it is sufficient to construct $O(1)$ such polynomials (see [5] for detailed analysis). Lemma 6 can be used to detect the convergence moment.

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

$$M_{\rho_1+\rho_2,r_1+r_2} = [S_i(x,y,z)P_j(x,y,z), i=0..\rho_1, j=0..\rho_2]. \tag{8}$$

Proof: See [5, Lemma 7]. ■

```
REDUCE($(S_0(x,y),\ldots,S_{i-1}(x,y)), P(x,y)$)
  $S_i(x,y) \leftarrow P(x,y)$
  while $\exists j : (0\leq j < i) \wedge (\operatorname{ydeg} S_j(x,y) = \operatorname{ydeg} S_i(x,y))$
  do if $\operatorname{LT} S_i(x,y) | \operatorname{LT} S_j(x,y)$
        then $W(x,y) \leftarrow S_j(x,y) - \frac{\operatorname{LT} S_j(x,y)}{\operatorname{LT} S_i(x,y)} S_i(x,y)$
             $S_j(x,y) \leftarrow S_i(x,y)$
             $S_i(x,y) \leftarrow W(x,y)$
        else $S_i(x,y) \leftarrow S_i(x,y) - \frac{\operatorname{LT} S_i(x,y)}{\operatorname{LT} S_j(x,y)} S_j(x,y)$
  if $S_i(x,y) = 0$
    then $i \leftarrow i-1$
  return $(S_0(x,y),\ldots,S_i(x,y))$
```

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

```
INTERPOLATE($q_{10}(x), q_{11}(x), \phi(x), r, \rho$)
  $\mathcal{G} \leftarrow (z\phi(x), y\phi(x))$
  $\mathcal{G} \leftarrow$ REDUCE($\mathcal{G}, zq_{11}(x) - yq_{10}(x)$))
  $\pi \leftarrow \lfloor \frac{\rho}{r} \rfloor$
  for $j \leftarrow 1$ to $\pi$
  do $\tilde{\mathcal{G}} = (z\mathcal{G}_0,\ldots,z\mathcal{G}_j, y\mathcal{G}_j)$
     while $\Delta(\tilde{\mathcal{G}}) > n$
     do $Q \leftarrow y\sum_{i=0}^{j} rand()\cdot\mathcal{G}_j$
        $\tilde{\mathcal{G}} \leftarrow$ REDUCE($\tilde{\mathcal{G}}, Q$))
     $\mathcal{G} \leftarrow \tilde{\mathcal{G}}$
  $\Pi = \pi$
  $\mathcal{B} \leftarrow \mathcal{G}$
   Let $r = \sum_{j=0}^{m} r_j 2^j, r_j \in \{0,1\}$
  $R \leftarrow 1$
  for $j \leftarrow m-1$ to $0$
  do $R \leftarrow 2R$
     $\Pi = 2\Pi$
     $\mathcal{B} \leftarrow$ MERGE($\mathcal{B}, \mathcal{B}, nR(R+1)/2$)
     if $r_j = 1$
       then $R \leftarrow R+1$
            $\Pi \leftarrow \Pi + \pi$
            $\mathcal{B} \leftarrow$ MERGE($\mathcal{B}, \mathcal{G}, nR(R+1)/2$)
  while $\Pi < \rho$
  do $\tilde{\mathcal{B}} = (z\mathcal{B}_0,\ldots,z\mathcal{B}_j, y\mathcal{B}_j)$
     while $\Delta(\tilde{\mathcal{B}}) > n\frac{r(r+1)}{2}$
     do $Q \leftarrow y\sum_{i=0}^{j} rand()\cdot\mathcal{B}_j$
        $\tilde{\mathcal{B}} \leftarrow$ REDUCE($\tilde{\mathcal{B}}, Q$))
     $\mathcal{B} \leftarrow \tilde{\mathcal{B}}$
     $\Pi \leftarrow \Pi + 1$
  return $\mathcal{B}$
```

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

This lemma allows one to generalize the binary interpolation algorithm proposed in [5] to the case of reformulated Wu list decoding method[1]. Namely, one can replace pairwise polynomial products in (8) 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

[1] The algorithm described here can be used in conjunction with the original Wu method as well.

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	0.83	1.78	0.48	0.11	11.7
Wu+binary	0.2	0.83	0.088	0.05	4.2
GS+re-encoding+binary	3.55	33	0.20	0.12	–

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 1 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 $M_{\rho_1+\rho_2,r_1+r_2}$ [5].

The described algorithm makes use of the function $Reduce$, which constructs a Gröbner basis of the module $\mathcal{M}' = \{S(x, y, z)+a(x)P(x, y, z)|S(x, y, z) \in \mathcal{M}\}$, given a Gröbner basis $(S_0(x, y, z), \dots, S_{i-1}(x, y, z))$ of some other module $\mathcal{M}$, and a partially homogenized polynomial $P(x, y, z)$. This function is shown in Figure 2. It can be considered as a multi-dimensional generalization of the extended Euclidean algorithm.

The proposed interpolation algorithm is summarized in Figure 3. $(1, w_1, w_2)$-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. Lemma 7 together with the randomized convergence speedup method are used on lines 3–9 to obtain from it a Gröbner basis of $M_{\pi,1}$, where π is selected to ensure that the subsequent steps would lead to a basis of $M_{\Pi,r}$ with Π as close as possible to ρ. The same approach is used on lines 22–28 to obtain a Gröbner basis of $M_{\rho,r}$ from the one of $M_{r\lfloor\frac{\rho}{r}\rfloor,r}$. Binary exponentiation algorithm is utilized on lines 12–21 to obtain a Gröbner basis of $M_{\pi r,r}$.

Let $S(x, y, z)$ be the smallest element of the basis produced by this algorithm. One should find all pairs $(a^{(j)}(x), b^{(j)}(x))$: $S(x, a^{(j)}(x), b^{(j)}(x)) = 0$ (see [6] for a generalization of the Roth-Ruckenstein algorithm to this case), and recover the corresponding error locator polynomials as $\sigma^{(j)}(x) = a^{(j)}(x)q_{10}(x) + b^{(j)}(x)q_{11}(x)$.

The most computationally intensive part of the proposed method is the multi-dimensional Euclidean algorithm ($Reduce$). Its complexity can be reduced by employing the generalization of Knuth-Schönhage algorithm given in [8].

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[2] 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 two. 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 a simple derivation of the Wu list decoding method 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, 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 two 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

This work was partially supported by the grant MK-1195.2009.9 of the President of Russia for young Candidates of Science.

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

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.

[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.

[5] P. Trifonov, “Efficient interpolation in the Guruswami-Sudan algorithm,” *IEEE Transactions on Information Theory, accepted for publication*, 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] 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.

[8] 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.