Improving erasure-resilient encoding/decoding by combining Cauchy and Hankel/Toeplitz matrix structure

Victor Y. Pan*
Mathematics and Computer Science Department
Lehman College, CUNY, Bronx, NY 10468
internet: vpan@alpha.lehman.cuny.edu

Abstract

We combine Cauchy and Toeplitz/Hankel matrix structure to improve the estimated running time and memory space of the known practical schemes for erasure-resilient encoding/decoding. Both encoding and decoding stages can be performed in nearly linear time by involving polynomial interpolation and multipoint evaluation. Due to the large overhead constants of the asymptotically fast known algorithms for the latter tasks (particularly, for interpolation), the resulting encoding/decoding scheme is, however, considered practically inferior to the more straightforward quadratic time algorithms. We propose a new algorithm (with its three variations at the encoding stage) that substantially decreases the latter overhead constants at both encoding and decoding stages. At the encoding stage, the algorithm also yields acceleration by logarithmic factor – to reach the complexity level of FFT or polynomial multiplication. Our computations do not involve polynomial interpolation. Multipoint polynomial evaluation is kept local to only one substage of the decoding computations. Even if this substage is replaced by the known algorithms running in quadratic time, our improvements at the other stages makes our algorithm practically very promising. Technically, we achieve our progress by combining the Cauchy structure of the generator matrix with Hankel/Toeplitz matrix structure.

Key words: erasure-resilient encoding/decoding, Cauchy matrices, Hankel/Toeplitz matrices.

1 Introduction

The power of using matrix structure for encoding/decoding algorithms is well known but was not fully explored yet. In this paper we exploit such a power to improve the computations by the known scheme for erasure-resilient encoding/decoding, at both encoding and decoding stages of the computations.

The known approach reduces the encoding to Trummer's celebrated problem (cf. [Go85], [PACLS98]) of multiplication of an $(n-m) \times m$ nonsingular Cauchy matrix C by a vector

*Supported by NSF Grant CCR 9732206 and PSC-CUNY Award 669363

provided that m information packets and $n-m$ redundant packets are sent through the communication channel and a total of $n-m$ packets are erased. The decoding is reduced to solving a nonsingular linear system of k equations whose coefficient matrix is a submatrix of C and, therefore, is also a Cauchy matrix (cf. [MS77], [ABELS94], [WB94], [AEL95] on this approach). Applications of the fast algorithms of [Ge87] for Trummer's problem and [Ga60] for solving a Cauchy linear system yield encoding/decoding at the computational cost $O(E(\ell)n/\ell + I(k) + E(k))$. Here and hereafter, $\ell = \min\{m, n-m\}$, and we write $E(s) = O(M(s)\log s)$, $I(s) = O(M(s)\log s)$, and $M(s) = O((s\log s)\log\log s)$ to denote the computational cost of the evaluation of a polynomial of degree s at $s+1$ points, interpolation to such a polynomial, and multiplication of two polynomials of degree s, respectively (cf. [BP94]). (We measure the computational cost in terms of the number of field operations involved and will refer to them as *ops*.)

The overhead constants hidden in the above "O" notation are quite large, however, particularly for the interpolation, and in practice the computation goes by more straightforward algorithms that involve $O((n-m)m)$ and $O(k^2)$ ops at the stages of encoding and decoding, respectively [BKKKLZ95].

We propose an alternative solution algorithm (with its three variations at the encoding stage), which better exploits the Cauchy structure involved. This enables us to avoid interpolation completely and to limit multipoint polynomial evaluation to the decoding stage. The encoding stage can be performed in $O(M(\ell)n/\ell)$ ops (see section 5) or, alternatively, by using $O(r)+2DFT(r)$ ops provided that $r \geq \max\{m, n-m\}$ and the ground field allows us to perform the discrete Fourier transform at the r-th roots of 1 by using *DFT(r)* ops, which is $O(r\log r)$ if FFT can be applied (see section 9). At the stage of decoding it is sufficient to use $O(E(k)+M(k))$ ops. In all our estimates, at both encoding and decoding stages, we achieve substantial decrease of the overhead constants in the "O" notation, versus the approach based on [Ga60] and [Ge87], and also save logarithmic factor in the time bound for encoding (cf. sections 6 and 10 and Remarks 7.1 and 8.1 in sections 7 and 8). All our computations are performed in linear space, of n words of memory at the encoding stage and from $O(k)$ to n at the decoding stage. The new techniques demonstrate the potential power of the application of structured matrices to encoding/decoding and promise to be practically useful for the erasure–resilient encoding/decoding. The algorithm allows its further practical improvements by using the XOR's computer words and tabulation (cf. [BKKKLZ95]). Our computations are reduced to polynomial nultiplication or FFT and, therefore, allow processor efficient parallel acceleration to yield the time bounds $O(\log m)$ and $O(\log^2 k)$ at the encoding and decoding stages, respectively.

Due to the well known importance of Trummer's problem [PACLS98], its solution proposed in section 7 may be of some independent interest.

Technically, our progress is achieved based on combining the Cauchy matrix structure, which ensures nonsingularity, with Hankel/Toeplitz structure, which supports additional theoretical and practical speed up of the computations.

We organize our paper as follows. After some definitions and preliminaries in the next section, we describe the basic encoding/decoding scheme in section 3, state our complexity estimates in section 4, and define the Hankel/Toeplitz matrix structure, which we combine with Cauchy structure, in section 5. In section 6 we study the inversion of a Cauchy matrix and summarize the resulting complexity estimates for our first algorithm, except for the stage of multiplication of the resulting inverse by a vector, which we cover in section 7. We describe two alternative algorithms in sections 8 and 9. Section 10 is left for a brief discussion.

Acknowledgement. The problem of improving the erasure-resilient encoding/decoding was brought to my attention by Marek Karpinski.

2 Definitions and some preliminary results

Fix two integers m and n, $n > m > 0$, a field $\mathbf{F}$ (say, $GF[2]$ or $GF[2^L]$) and its n distinct elements s_i, $i = 0, \ldots, n-m-1$, and $-t_j$, $j = 0, \ldots, m-1$. Let I_s stand for the $s \times s$ identity matrix. Let $\mathbf{s} = (s_i)_{i=0}^{n-m-1} \in \mathbf{F}^{n-m}$, $\mathbf{t} = (t_j)_{j=0}^{m-1} \in \mathbf{F}^m$, $\mathbf{u} = (u_i)_{i=0}^{p-1} \in \mathbf{F}^p$, $\mathbf{v} = (v_i)_{i=0}^{q-1} \in \mathbf{F}^q$ be some vectors of dimensions $n-m, m, p$ and q, resp. For a vector $\mathbf{r} = (r_i)_{i=0}^{k-1}$, let $V(\mathbf{r}) = (r_i^j)_{i,j=0}^{k-1}$ be a $k \times k$ Vandermonde matrix. For a vector or a matrix W, let W^T be its transpose. Let $C = C(\mathbf{u}, \mathbf{v}) = (\frac{1}{u_i+v_j})_{i=0,j=0}^{p-1,q-1}$ be a $p \times q$ *Cauchy matrix*, defined by two vectors $\mathbf{u}$ and $\mathbf{v}$, and let $G = G(\mathbf{u}, \mathbf{v}) = \begin{pmatrix} I_m \\ C(\mathbf{u}, \mathbf{v}) \end{pmatrix} \in \mathbf{F}^{n \times m}$ be the $n \times m$ *generator matrix.* diag $(\mathbf{u}) = \mathrm{diag}(u_i)_{i=0}^{p-1}$ will be our notation for the $p \times p$ diagonal matrix with diagonal entries $u_0, \ldots, u_{p-1}$; $h_{\mathbf{u}}(x)$ for the monic polynomial $\prod_{i=0}^{p-1}(x - u_i)$ of degree p in x, $h'_{\mathbf{u}}(x)$ for its derivative in x, and card (S) for the cardinality of a set S.

For a vector $\mathbf{v} = (v_j)_{j=0}^{q-1}$, a matrix $W = (w_{j,k})_{j=0,k=0}^{q-1,p-1}$, and two subsets, J of the set $\{0, \ldots, q-1\}$ and K of the set $\{0, \ldots, p-1\}$, let $\mathbf{v}_J$ be the subvector $(v_j)_{j \in J}$ of $\mathbf{v}$ and let $W_{J,K}$ be the submatrix $(w_{j,k})_{j \in J, k \in K}$ of W.

Hereafter, "ops" stands for "operations in the field $\mathbf{F}$", $M(s)$ ops suffice to multiply modulo x^s a pair of polynomials in x,

$$M(s) = O((s \log s) \log \log s), \tag{2.1}$$

$H(k)$ ops suffice to compute the coefficients of the polynomial $h_{\mathbf{u}}(x)$ for a given vector $\mathbf{u} \in \mathbf{F}^k$,

$$H(k) = O(M(k) \log k), \tag{2.2}$$

$E(k)$ ops suffice to evaluate a polynomial of degree at most $k-1$ on a set of k points,

$$E(k) = O(M(k) \log k), \tag{2.3}$$

and $DFT(k)$ ops suffice to perform discrete Fourier transform on the set of the k-th roots of 1,

$$DFT(k) = O(k \log k) \tag{2.4}$$

over a field that supports FFT (cf. e.g. [BP94] on the proofs of (2.1)-(2.4)).

Remark 2.1 *In the field $\mathbf{F} = GF(2^L)$, the ops can be performed as operations with polynomials modulo a fixed irreducible polynomial of degree L or as XOR's of computer words [MS77], [BKKKLZ95], XOR's being the better choice in practice.*

The minimum number of ops, required to multiply a fixed rectangular or square matrix W by a vector will be denoted by c_W.

Lemma 2.1 *$c_V \leq c_W$ for any submatrix V of a matrix W.*

Lemma 2.2 *Let V be a $k \times \ell$ block matrix with $g \times h$ blocks $V(i,j)$, $i = 0, \ldots, k-1$; $j = 0, \ldots, \ell-1$. Then $c_V \leq \sum_{i,j} c_{V(i,j)} + (\ell - 1)gk$.*

We will also use the following results, the first and the third of which are obvious:

Lemma 2.3 $c_{V(\mathbf{r})} = E(k)$.

Lemma 2.4 *(cf. [PSD70]).* $c_{V^T(\mathbf{r})} = c_{V(\mathbf{r})}$.

Lemma 2.5 *If the field* $\mathbf{F}$ *supports FFT on the set of the* k*-th roots of 1 and if* $\mathbf{r}$ *is a vector of the* k*-th roots of 1, then* $C_{V(\mathbf{r})} = DFT(k)$.

Remark 2.2 *By Lemma 2.1, the estimates of Lemmas 2.3-2.5 apply to all rectangular submatrices of* $V(\mathbf{r})$ *and* $V^T(\mathbf{r})$.

Hereafter, we will write

$$\ell = \min\{m, n-m\}. \tag{2.5}$$

3 Erasure-resilient encoding/decoding scheme

The next scheme for erasure-resilient encoding/decoding involves a Cauchy matrix and relies on a Reed-Solomon systematic linear code with an m-packet message vector $\mathbf{v}$ and a generator matrix $G = G(\mathbf{s}, \mathbf{t})$ [MS77], [ABELS94], [WB94], [AEL95]. Stage 1 represents encoding, stages 2-5 decoding. Practically, the scheme is applied to several vectors $\mathbf{v}$, which are processed successively or concurrently.
Algorithm 3.1 *(encoding/decoding).*

1. (*encoding* m*-packet message* $\mathbf{v}$). Compute the vector $\mathbf{u} = G\mathbf{v}$.
2. Define a subset Q of $\text{card}(Q) = n - m$ in the set $\{0, \dots, n-1\}$, (Q indexes the $n - m$ erased packets); its intersections, K, with the set $\{0, \dots, m-1\}$, and P, with the set $\{m, \dots, n-1\}$; the complements of these intersections, $M = \{0, \dots, m-1\} - K$ and $J = \{m, \dots, n-1\} - P$; the subvectors $\mathbf{u}_K$ and $\mathbf{u}_M$ of $\mathbf{u}$, and the square Cauchy matrix $C_{J,K} \in \mathbf{F}^{k \times k}$ (a submatrix of $C = C(\mathbf{s}, \mathbf{t})$, for $k = \text{card}(J) = \text{card}(K)$).
3. Compute the matrix $\tilde{C} = C_{J,K}^{-1}$.
4. Compute the vector $\mathbf{v}_K = \tilde{C}\mathbf{u}_K$.
5. Output the vector $\mathbf{v}$ composed of its subvectors $\mathbf{v}_M = \mathbf{u}_M$ and $\mathbf{v}_K$.

The use of Cauchy matrix C is due to the next theorem, which also implies the correctness of Algorithms 3.1 [MS77].

Theorem 3.1 *(cf. e.g. [M82]). Every square submatrix of the matrix* $C = (\frac{1}{s_i + t_j})_{i,j}$ *is nonsingular if all* n *values* $s_i, -t_j$ *are distinct.*

4 Computational cost estimates

No ops are used at Stages 2 and 5 of Algorithm 3.1. The asymptotic computational complexity at stages 1, 3 and 4 can be estimated by using the known algorithms of [Ga60] and [Ge87] for computations with Cauchy matrices. In particular, due to the structure of a Cauchy matrix C, it can be stored economically (if we represent it by n parameters s_i, t_j) and multiplied by a vector fast (cf. [Ge87] or [BP94], p.130):

Proposition 4.1 *A $k \times k$ Cauchy matrix can be multiplied by a vector in $O(M(k)\log k)$ ops (for $M(k)$ of (2.1)).*

Furthermore, we have (cf. [Ga60] or [BP94], p. 131):

Proposition 4.2 *$O(M(n)\log k)$ ops suffice to solve a linear system of k equations with a $k \times k$ Cauchy coefficient matrix.*

The computation of the vectors $\mathbf{u}$ at stage 1 and $\mathbf{v}$ at stages 3 and 4 of Algorithm 3.1 requires only $O(M(\ell)(\log \ell)n/\ell) = O((n\log^2 \ell)\log\log \ell)$ and $O(M(k)\log k) = O((k\log^2 k)\log\log k)$ ops, respectively (cf. (2.1),(2.5)). Practically, the users prefer asymptotically slower algorithms that require ($O((n-m)m)$, $O(k^2)$, and $O(k^2)$ ops at stages 1, 3 and 4, respectively (cf. [BKKKLZ95]). This is because the algorithms of [Ga60], [Ge87] rely on asymptotically fast polynomial interpolation and multipoint evaluation, which involve (particularly the interpolation) larger overhead constants. Our algorithm supporting the next theorem (where we use relations (2.1)-(2.5)) improves the cited asymptotically fast computations by reducing the overhead constants substantially and by accelerating encoding by logarithmic factor (cf. also Remarks 7.1 and 8.1 in sections 7 and 8).

Theorem 4.1 *Algorithm 3.1 can be performed by using $O((n/\ell)M(\ell))$ ops at stage 1 (for ℓ of (2.5)), $O(H(k)+E(k))$ at stage 3, and $O(E(k)+M(k))$ at stage 4 provided that the matrix $\tilde{C}$ is represented in factorized form with $4k$ parameters (rather than with its k^2 entries), and similarly the input matrix C is represented by the n parameters s_i, t_j. At all stages, the computation requires only linear memory space.*

Our first observation towards the proof of Theorem 4.1 is that in the encoding/decoding process the integers m and n are given by users, the set Q is defined by the uncontrolled erasure process in the transmission of the message, and we may choose any vectors $\mathbf{s}$ and $\mathbf{t}$. We will use this choice to impose some additional structure on the Cauchy matrix C.

5 Cauchy matrices with Hankel/Toeplitz structrure

$H = (h_{i,j})$ (resp. $T = (t_{i,j})$) is a *Hankel* (resp. *Toeplitz*) matrix if $h_{i+1,j-1} = h_{i,j}$ (resp. $t_{i+1,j+1} = t_{i,j}$) for all i,j for which $h_{i+1,j-1}$ and $h_{i,j}$ (resp. $t_{i+1,j+1}$ and $t_{i,j}$) are defined.

Fact 5.1 *H is a Hankel matrix, and T is a Toeplitz matrix if and only if so are their transposes.*

Theorem 5.1 *(cf. [BP94] and Lemma 2.2). $O(M(l)n/l)$ ops and $O(n)$ memory space suffice to multiply an $(n-m)\times m$ Hankel or Toeplitz matrix by a vector, for ℓ of (2.5).*

We immediately observe

Fact 5.2 *A Cauchy matrix $C = (\frac{1}{s_i - t_j})$ is a Hankel matrix if and only if $s_{i+1} - s_i = t_j - t_{j+1}$ for all pairs (i,j) for which s_{i+1}, s_i, t_{j+1}, and t_j are defined. C is a Toeplitz matrix if and only if $s_{i+1} - s_i = t_{j+1} - t_j$ for all pairs (i,j) for which s_{i+1}, s_i, t_{j+1}, and t_j are defined.*

In particular, $C = (\frac{1}{i+j+a})_{i,j}$ is a Hankel matrix, $C = (\frac{1}{i-j+a})_{i,j}$ is a Toeplitz matrix, for any scalar a such that $i+j+a \neq 0$ (or resp. $i-j+a \neq 0$) for all pairs (i,j) from the allowed range.

By choosing a Cauchy matrix C of this kind as our input to Algorithm 3.1, we achieve the desired time and space complexity bounds at stage 1.

6 The inverse of a Cauchy matrix

Theorem 6.1 *(cf. e.g. [R89] or [BP94], p. 131). Let $\mathbf{p} = (p_i)_{i=0}^{k-1}$, $\mathbf{q} = (q_i)_{i=0}^{k-1}$, where $p_0, q_0, \dots, p_{k-1}, q_{k-1}$ are distinct. Then $\det C(\mathbf{p}, \mathbf{q}) \neq 0$ and $C^{-1}(\mathbf{p}, \mathbf{q}) = D_0 C(\mathbf{q}, \mathbf{p}) D_1$, where*
$D_0 = \operatorname{diag}((-1)^i \frac{h_{\mathbf{p}}(q_i)}{h'_{\mathbf{q}}(q_i)})_{i=0}^{k-1}$, $D_1 = \operatorname{diag}((-1)^j \frac{h_{\mathbf{q}}(p_j)}{h'_{\mathbf{p}}(p_j)})_{j=0}^{k-1}$.

Theorem 6.2 *(cf. e.g. [BP94]). Given two vectors $\mathbf{p}, \mathbf{q} \in \mathbf{F}^k$, it suffices to use $O(H(k) + E(k)) = O(M(k) \log k)$ (for $M(k), H(k)$ and $E(k)$ of (2.1)-(2.3)) ops to compute the coefficients of the polynomials $h_{\mathbf{p}}(x)$, $h_{\mathbf{q}}(x)$, $h'_{\mathbf{p}}(x)$, and $h'_{\mathbf{q}}(x)$ and the vectors $((-1)^i h_{\mathbf{p}}(q_i)/h'_{\mathbf{q}}(q_i))_{i=0}^{k-1}$ and $((-1)^j h_{\mathbf{q}}(p_j)/h'_{\mathbf{p}}(p_j))_{j=0}^{k-1}$, defining the matrices D_0 and D_1 of Theorem 6.1.*

We also observe the following simple facts:

Fact 6.1 *If $C(\mathbf{p}, \mathbf{q}) = (C(\mathbf{s}, \mathbf{t}))_{K,J}$, then $C(\mathbf{q}, \mathbf{p}) = (C(\mathbf{t}, \mathbf{s}))_{J,K}$.*

Fact 6.2 *$C(\mathbf{t}, \mathbf{s}) = -C^T(\mathbf{s}, \mathbf{t})$.*

By combining Theorems 5.1, 5.2, 6.1, 6.2, Proposition 4.1, and Facts 5.1, 5.2, 6.1 and 6.2, we obtain the desired complexity bounds at stages 1 and 3 of Algorithm 3.1 (cf. the end of the previous section).

Corollary 6.1 *Let vectors $\mathbf{s}, \mathbf{t}$ satisfy the assumptions of Fact 5.3, that is, let the Cauchy matrix $C = C(\mathbf{s}, \mathbf{t})$ be also a Hankel (or Toeplitz) matrix. Then Stage 1 of Algorithm 3.1 involves $O(n)$ memory space and $O(H(k) + E(k)) = O(M(\ell) n/\ell)$ ops for $M(\ell)$ of (2.1) and $\ell = \min\{m, n-m\}$, whereas Stage 3 requires $O(k)$ memory space and $O(H(k) + E(k)) = O(M(k) \log k)$ ops to compute the diagonal matrices D_0, D_1 of Theorem 6.1, defining the matrix $C^{-1}(\mathbf{p}, \mathbf{q})$ in the factorized form $D_0 C(\mathbf{q}, \mathbf{p}) D_1$, where $\mathbf{p} = \mathbf{s}_J$, $\mathbf{q} = \mathbf{t}_K$. Furthermore, the matrix $C(\mathbf{q}, \mathbf{p}) = (C(\mathbf{s}, \mathbf{t}))_{K,J}$ is a submatrix of the $(n-m) \times m$ Cauchy and Hankel (or Toeplitz) matrix $C(\mathbf{s}, \mathbf{t})$.*

7 Computational cost of multiplication of the inverse of a Cauchy matrix by a vector

The $k \times k$ diagonal matrices D_0 and D_1 are multiplied by a vector in $2k$ ops. Therefore, the cost of multiplication by a vector of the $k \times k$ matrix $E = C^{-1}(\mathbf{p}, \mathbf{q}) = D_0 C(\mathbf{q}, \mathbf{p}) D_1$, for $\mathbf{p} = \mathbf{s}_J$, $\mathbf{q} = \mathbf{t}_K$, is dominated by the cost of multiplication of the $k \times k$ Cauchy matrix $C(\mathbf{q}, \mathbf{p})$ by a vector (Trummer's problem). We propose to solve this problem based on the following factorization:

Fact 7.1 *[FHR93]. $C(\mathbf{q}, \mathbf{p}) = (\mathrm{diag}(h_{\mathbf{p}}(q_i))_{i=0}^{k-1})^{-1} V(\mathbf{q}) H(\mathbf{p}) V^T(\mathbf{p})$, where $H(\mathbf{p}) = (h_{i,j})$ is a triangular Hankel matrix, $h_{i,j} = p_{i+j}$ where $i + j < k$, $h_{i,j} = 0$ where $i + j \geq k$, $\mathbf{p} = (p_i)_{i=0}^{k-1}$.*

We observe that the values $h_{\mathbf{p}}(q_i)$, $i = 0, \ldots, k-1$, are available from stage 3 (actually, the factor $(\mathrm{diag}(h_{\mathbf{p}}(q_i))_i)^{-1}$ of Fact 7.1 and the numerator-diagonal of D_0 cancel each other). By combining Lemmas 2.3 and 2.4 with Theorem 5.1, we obtain the solution of Trummer's problem of multipication of $C(\mathbf{q}, \mathbf{p})$ by a vector by using $O(E(k) + M(k)) = O(M(k) \log k)$ ops. This completes the proof of Theorem 4.1. □

Remark 7.1 *An alternative way to fast decoding is to embed the matrix $C(\mathbf{q}, \mathbf{p})$ into $C(\mathbf{t}, \mathbf{s})$ and then to apply Theorem 5.1 and Lemma 2.1. Such a solution of Trummer's problem for $C(\mathbf{q}, \mathbf{p})$ involves $O(M(l)n/l)$ ops (as at the encoding stage), which may improve the estimate $O(H(k) + E(k))$ of Theorem 4.1, depending on how large k is versus n. The same observations apply to the algorithms of the next two sections.*

8 An alternative choice of a Cauchy matrix

If the field $\mathbf{F}$ supports FFT on the set of the r-th roots of 1 for $r \geq \max\{m, n-m\}$, we may improve encoding by assigning, say, $s_i = a\omega_r^i$, $i = 0, \ldots, n-m-1$; $t_j = b\omega_r^{n-m+j}$, $j = 0, \ldots, m-1$, where ω_r is a primitive r-th root of 1, and the scalars a and b are chosen to insure that the n values as_i and bt_j are all distinct. In this case, we factorize the Cauchy matrix $C(\mathbf{s}, \mathbf{t})$ based on Fact 7.1, compute or *pre-compute* the coefficients of the polynomial $h_{\mathbf{t}}(x) = \prod_{j=0}^{m-1}(x - a\omega_r^{n-m+j})$ (at the cost at most $H(m)$), then its values $h_{\mathbf{t}}(s_i)$, $i = 0, \ldots, n-m-1$ (by applying FFT), and finally multiply $C(\mathbf{s}, \mathbf{t})$ by a vector $\mathbf{v}$ based on Lemma 2.5, Remark 2.1 and Theorem 5.1.

Remark 8.1 *If we embed the matrices $V(\mathbf{q})$ and/or $V^T(\mathbf{p})$ into $V(\mathbf{t})$ and/or $V^T(\mathbf{s})$, respectively, and then apply Lemma 2.5 and Remark 2.1, then the multiplication of $V(\mathbf{q})$ and/or $V^T(\mathbf{p})$ by a vector will involve $O(M(m))$ and/or $O(M(n-m))$ ops, respectively, which may improve the bound $O(M(k) \log k)$, depending on how large k is versus m and/or $n-m$.*

9 A variant of encoding

Assuming that the field $\mathbf{F}$ contains the r-th roots of 1, for $r \geq \max\{m, n-m\}$, we may try to simplify further the encoding stage, by reducing it essentially to two discrete Fourier transforms on the set of the r-th roots of 1.

We will start with some definitions. We write $\mathbf{w} = (\omega_r^i)_{i=0}^{r-1}$, where ω_r is a primitive r-th root of 1; $\Omega = (\omega^{ij})_{i,j=0}^{r-1}$, Ω denoting the matrix of DFT.

For a scalar e, we define the $r \times r$ unit e-circulant matrix, $T_e = (t_{i,j})_{i,j=0}^{r-1}$, $t_{i,j} = 1$ for $i = j+1$, $t_{i,j} = e$ for $i = 0$, $j = r-1$, $t_{i,j} = 0$ for other i, j. The matrix T_e generates the class of e-circulant matrices, $T_e(\mathbf{v}) = \sum_{i=0}^{r-1} v_i T_e^i$, for all vectors $\mathbf{v} = (v_i)_{i=0}^{r-1}$. (This is a subclass of the class of Toeplitz matrices.) We write $D(\mathbf{v}) = \mathrm{diag}(v_i)_{i=0}^{r-1}$.

Theorem 9.1 *Let $g, h, e = g^r, f = h^r$ be four nonzero scalars, $e \neq f$, let $\mathbf{e} = (g^i)_{i=0}^{r-1}$, $\mathbf{f} = (h^i)_{i=0}^{r-1}$, $U_e = \Omega D(\mathbf{e}), U_f = \Omega D(\mathbf{f})$, and let the vector $U_f(\mathbf{v})$ (for a given vector $\mathbf{v}$) have no zero components. Furthermore, let the vectors $g\mathbf{w}$ and $h\mathbf{w}$ have $2r$ distinct components, that is, let the Cauchy matrix $\hat{C} = C(g\mathbf{w}, h\mathbf{w})$ be well defined and nonsigular. Then* $\hat{C} = \frac{1}{e-f} D^{-1}(U_f(\mathbf{v}))U_f T_e(\mathbf{v}) U_e^{-1} D^{-1}(U_e^{-1})^T \mathbf{e}^{(r-1)})$.

Proof. Theorem 9.1 is deduced from Proposition 6.3 of [P99]. □

Now, let $r \geq \max\{m, n-m\}$, fix two nonzero scalars, g and h, and a vector $\mathbf{v}$ (say, $\mathbf{v} = \mathbf{e}^{(1)}$) satisfying the assumptions of Theorem 9.1, and define the matrix $C(\mathbf{s}, \mathbf{t}) \in \mathbf{F}^{(n-m)\times m}$ being a block of the matrix $\hat{C}$ of (9.1). In this case, Trummer's problem of multiplication of $\hat{C}$ (and hence also of $C(\mathbf{s}, \mathbf{t})$) by a vector is reduced essentially (up to performing $O(n)$ ops) to multiplication of the matrices Ω and $(\Omega^{-1})^T$ by a vector, that is, to two discrete Fourier transforms on the set of the r-th roots of 1, because the choice of $\mathbf{v} = \mathbf{e}^{(i)}$ for any i from 1 to $r-1$ trivializes the computation of the vector $U_f\mathbf{v}$ and the multiplication of $T_e(\mathbf{v})$ by a vector. Actually, the computations are further simplified because we only multiply the submatrix $C(\mathbf{s}, \mathbf{t})$ of $\hat{C}$ (rather than $\hat{C}$ itself) by a vector. Further minor simplification is yielded for $e = 1$ or $f = 1$ because $D(\mathbf{1}) = I_r$ and, therefore, $U_1 = \Omega$.

10 Discussion

Various further modifications of the proposed approach are possible and may lead to some further improvements. For instance, one may try to improve the decoding stage based on the following expressions:

$$C^{-1}(\mathbf{p}, \mathbf{q}) = (\mathrm{diag}(h'_{\mathbf{q}}(q_i))_{i=0}^{k-1})^{-1}\ V(\mathbf{q})V^{-1}(\mathbf{p})\mathrm{diag}(h_{\mathbf{q}}(p_i))_{i=0}^{k-1},$$

$$V^{-1}(\mathbf{p}) = RT_f(\mathbf{h_p} + f\mathbf{e}^{(0)})V^T(\mathbf{p})(\mathrm{diag}(h'_{\mathbf{p}}(p_i)(f - p_i^n))_{i=0}^{k-1})^{-1},$$

where f is a scalar, $p_i^n \neq f$ for $i = 0, \ldots, k-1$, $\mathbf{h_p} = (h_i)_{i=0}^{k-1}$ for $h_0, \ldots, h_{k-1}$ satisfying the equation $x^k + \sum_{i=0}^{k} h_i x^i = \prod_{i=0}^{k-1}(x - p_i)$, $\mathbf{e}^{(0)} = (1, 0, \ldots, 0)^T$, and R is the reversion matrix, $R = (r_{i,j})$, $r_{i,j} = 1$ for $i = k-1-j$, $r_{i,j} = 0$ otherwise (cf. [FHR93], [BP94], p.144). In this case, one may multiply $V^T(\mathbf{p})$ by a vector based on Tellegen's theorem (cf. Lemmas 2.3 and 2.4).

References

[ABELS94] A. Albanese, J. Blömer, J. Edmonds, M. Luby, M. Sudan, Priority Encoding Transmission, *Proc. 35th Ann. Symp. Foundations of Computer Science (FOCS)*, 604-613, IEEE Computer Society Press, 1994.

[AEL95] N. Alon, J. Edmonds, M. Luby, Linear Time Erasure Codes with Nearly Optimal Recovery, *Proc. 36th Ann. Symp. Foundations of Computer Science (FOCS)*, IEEE Computer Society Press, 1995.

[BP94] D. Bini, V.Y. Pan, *Polynomial and Matrix Computations, Volume 1: Fundamental Algorithms*, Birkhäuser, Boston, 1994.

[BKKKLZ95] J. Blömer, M. Kalfane, R. Karp, M. Karpinski, M. Luby, D. Zuckerman, An XOR-Based Erasure-Resilient Coding Scheme Technical Report TR-95-48, *International Computer Science Institute*, Berkeley, Califormia, 1995.

[FHR93] T. Fink, G. Heinig, K. Rost, An Inversion Formula and Fast Algorithms for Cauchy-Vandermonde Matrices, *Linear Algebra Appl.*, **183**, 179–191, 1993.

[Ga60] N. Gastinel, Inversion d'une Matrice Generalisant la Matrice de Hilbert, *Chiffres*, 3, 149-152, 1960.

[Ge87] A. Gerasoulis, A Fast Algorithm for the Multiplication of Generalized Hilbert Matrices with Vectors, *Math. Comp.*, **50, 181**, 179-188, 1987.

[Go85] G. H. Golub, Trummer's Problem, *SIGACT News*, **17, 2**, 12, ACM Press, New York, 1985.

[M82] L. Mirsky, *An Introduction to Linear Algebra*, Dover, New York, 1982.

[MS77] F. J. MacWilliams, N. J. A. Sloane, *The Theory of Error-Correcting Codes*, North-Holland, New York, 1977.

[P90] V. Y. Pan, Computations with Dense Structured Matrices, *Math. of Computation*, **55**, **191**, 179–190, 1990.

[P99] V. Y. Pan, Nearly Optimal Computations with Structured Matrices, preprint, 1999.

[PACLS98] V. Y. Pan, M. AbuTabanjeh, Z. Chen, E. Landowne, A. Sadikou, New Transformations of Cauchy Matrices and Trummer's Problem, *Computer and Math. (with Applics.)*, **35, 12**, 1-5, 1998.

[PSD70] P. Penfield Jr., R. Spencer, S. Duinker, *Tellegen's Theorem and Electrical Networks,* MIT Press, Cambridge, Massachusetts, 1970.

[R89] M. O. Rabin, Efficient Dispersal of Information for Security, Load Balancing, and Fault Tolerance, *J. ACM*, **36, 2**, 335-348, 1989.

[WB94] S. B. Wicker, V. Bhargava, *Reed-Solomon Codes and Their Applications*, IEEE Press, New York, 1994.