Gerchberg-Papoulis algorithm and the finite Zak transform

Andrzej K. Brodzik*,[a] and Richard Tolimieri[b]

[a]Scientific Software, Woburn, MA 01801
[b]Psypher Inc., Boston, MA 02134

ABSTRACT

We propose a new, time-frequency formulation of the Gerchberg-Papoulis algorithm for extrapolation of band-limited signals. The new formulation is obtained by translating the constituent operations of the Gerchberg-Papoulis procedure, the truncation and the Fourier transform, into the language of the finite Zak transform, a time-frequency tool intimately related to the Fourier transform. We will show that the use of the Zak transform results in a significant reduction of the computational complexity of the Gerchberg-Papoulis procedure and in an increased flexibility of the algorithm.

Keywords: Gerchberg-Papoulis algorithm, signal extrapolation, Zak transform

1. INTRODUCTION

The extrapolation of band-limited signals is encountered in many applications, including limited angle tomography, radio astronomy, synthetic aperture radar, geophysical exploration and communication theory. In each of these applications the problem is to recover the missing data in a one- or a multi-dimensional signal, given a signal that is band-limited in some way. A well known technique suitable for the extrapolation problem is the Gerchberg-Papoulis (GP) algorithm [6,9]. The algorithm is an iterative procedure in which one alternates between the object domain and the Fourier transform domain while constraining the signal to its known values and to a finite bandwidth. While this procedure is computationaly efficient and easy to implement (its core computation is a DFT), it converges slowly and often requires computation of several hundreds iterations. In this work we propose a new, Zak space formulation of the GP algorithm which significantly reduces computational complexity of the original approach.

Denote by $\mathcal{C}^N$ the N-dimensional space of N-tuples of complex numbers. A vector $f \in \mathcal{C}^N$ is a column vector

$$f = \begin{bmatrix} f_0 \\ \vdots \\ f_{N-1} \end{bmatrix}, \quad f_n \in \mathcal{C}, \quad 0 \le n < N.$$

Consider a partially known signal $f \in \mathcal{C}^N$. Typically only certain segments of f are known. Suppose $N = LM$ and segment f into L contiguous segments in $\mathcal{C}^M$

$$f = \begin{bmatrix} f_0 \\ \vdots \\ f_{L-1} \end{bmatrix}, \quad f_l \in \mathcal{C}^M, \quad 0 \le l < L. \tag{1}$$

*Correspondence: Email: brodzik@max.rl.plh.af.mil; Telephone: 781 377 4047; Fax: 781 377 4018.

For $0 < p \leq q \leq L$, suppose the known segments are

$$f_{p-1}, \ldots, f_{q-1}.$$

Denote by

$$\mathbf{f}_{n'} = F_N f_{n'} = \sum_{n=0}^{N-1} f_n e^{2\pi i n n'/N}, \quad 0 \leq n' < N,$$

the Fourier transform of f.

Suppose $\mathbf{f}$ is bandlimited. Typically only certain segments of $\mathbf{f}$ are nonzero while the remaining segments vanish. Segment $\mathbf{f}$ into M contiguous subvectors in $\mathcal{C}^L$

$$\mathbf{f} = \begin{bmatrix} \mathbf{f}_0 \\ \vdots \\ \mathbf{f}_{M-1} \end{bmatrix}, \quad \mathbf{f}_m \in \mathcal{C}^L, \quad 0 \leq m < M. \tag{2}$$

For $0 < p' \leq q' \leq M$, suppose the known segments are

$$\mathbf{f}_{p'-1}, \ldots, \mathbf{f}_{q'-1}.$$

The Gechberg-Papoulis algorithm is an iterative scheme for filling in the missing components of a signal f using as a constraint in the object domain the known parts of f and as a constraint in the Fourier domain the bandlimited information. Truncation operations play a major role in describing the procedure. Denote by $t_L^{p,q}$ the object domain linear operator which acts by the identity mapping on the segments $f_{p-1}, \ldots, f_{q-1}$ and by the zero mapping on the remaining segments in (1). In the same way, $t_M^{p',q'}$ is the Fourier domain operator which acts by the identity mapping on the segments $\mathbf{f}_{p'-1}, \ldots, \mathbf{f}_{q'-1}$ and by the zero mapping on the remaining segments in (2). The Gerchberg-Papoulis iteration is given by

$$f^{k+1} = f^0 + (I - t_L^{p,q}) F_N^{-1} t_M^{p',q'} F_N f^k, \tag{3}$$

where the initial estimate $f^0 = t_L^{p,q} f$ depends only on the known parts of f, F_N^{-1} denotes the inverse Fourier transform, and I is the identity operator. Equation (3) can be expressed compactly as

$$f^{k+1} = P_{SP} P_{BL} f^k,$$

where $P_{BL} = F_N^{-1} t_M^{p',q'} F_N$ and $P_{SP} = f^0 + (I - t_L^{p,q})$ are constraint operators enforcing finite bandwidth and known signal part.

The multiplicative complexity of a single iteration is $2N log_2 N$, due to the forward and inverse Fourier transforms. Since each iteration of the Gerchberg-Papoulis algorithm involves transition between the time and frequency domains, it is natural to ask if an improvement could be made by considering signal extrapolation in a joined time-frequency space. In this paper we consider a translation of the Gerchberg-Papoulis algorithm into the language of the finite Zak transform, a time-frequency tool intimately related to the Fourier transform. We will show that the finite Zak transform can be used to reduce the multiplicative complexity of the Gerchberg-Papoulis iteration, by up to a factor of $log_2 N$.

In the next section we will review the relevant properties of the finite Zak transform.

2. FINITE ZAK TRANSFORM

The finite Fourier transform $F_N f$ describes the frequency components of a signal $f \in \mathcal{C}^N$. The finite Zak transform provides a simultaneous encoding of signal and frequency information. It has its origin in several applications [15] and is usually the first stage in a time-frequency analysis including ambiguity functions [13], Gabor and Weyl-Heisenberg expansions [1,3,4,5,13], and wavelet transform [10].

Suppose L is a divisor of N and $N = LM$. For $f \in \mathcal{C}^N$ define the *finite Zak transform* (FZT) $Z_L f(a,b)$, by

$$Z_L f(a,b) = \sum_{r=0}^{L-1} f(a+rM) e^{2\pi i r b/L}, \quad 0 \le a < M, \quad 0 \le b < L. \tag{4}$$

If $L = 1$, the associated finite Zak transform is the identity mapping on $\mathcal{C}^N$ while if $L = N$, the associated finite Zak transform is the finite Fourier transform F_N. For an arbitrary divisor L of N, $Z_L f$ computes the finite Fourier transforms of the M vectors in $\mathcal{C}^L$ formed by striding by M through the signal $f \in \mathcal{C}^N$.

The FZT calculus is governed by several rules that allow manipulation of signals in the Zak space. We state several elementary results. Details can be found in [7].

The signal f can be recovered from its Zak transform by the inverse transform Z_L^{-1}

$$f(a+rM) = Z_L^{-1}(Z_L f(a,b)) = L^{-1} \sum_{b=0}^{L-1} Z_L f(a,b) e^{-2\pi i r b/L}, \quad 0 \le a < M, \quad 0 \le r < L. \tag{5}$$

A fundamental property of the finite Zak transform is the relation between $Z_L f$ and $Z_M \mathbf{f}$.

Theorem 1 *If $f \in \mathcal{C}^N$ and $\mathbf{f}$ is its N-point Fourier transform, then their Zak transforms $Z_L f(a,b)$ and $Z_M \mathbf{f}(b,a)$ are related as follows*

$$Z_M \mathbf{f}(a,b) = M Z_L f(-b,a) e^{-2\pi i a b/N}.$$

Proof: See [2].

Theorem 1 asserts that 90 degrees rotation of the Zak transform $Z_L f(a,b)$ and multiplication by the factor $e^{-2\pi i ab/N}$ produces the Zak transform $Z_M \mathbf{f}(a,b)$. This operation is equivalent to computing $(Z_M F_N Z_L^{-1})(Z_L f)$. Since the procedure is $logN$ less computationally expensive then the Fourier transform, algorithms requiring estimation of both: time and frequency properties of signals can be more conveniently implemented in the Zak space.

Truncation operations play a major role in many signal processing applications. Here we state a new result which relates Zak transforms of a signal and its truncation.

Take $f_r \in \mathcal{C}^M$ to be the r-th segment of $f \in \mathcal{C}^N$ and form a signal $f^{p,q}$ having zero segments as its first p segments and as its last $L-q$ segments, and equal to f at the remaining segments. The Zak transform of $f^{p,q}$ is

$$Z_L^{p,q} f(a,b) = \sum_{r=p-1}^{q-1} f(a+rM) e^{2\pi i r b/L}, \quad 0 \le a < M, \quad 0 \le b < L. \tag{6}$$

The following result describes the relation between $Z_L^{p,q} f$ and $Z_L f$.

Theorem 2 *If $Z_L^{p,q} f(a,b)$ is the Zak transform of $f_{p,q}$, and $Z_L f(a,b)$ is the Zak transform of f then*

$$Z_L^{p,q} f(a,b) = T_L^{p,q} Z_L f(a,b), \quad 0 \le a < M, \quad 0 \le b < L,$$

where $T_L^{p,q}$ is the Zak space trancation operator given by

$$T_L^{p,q} Z_L f(a,b) = L^{-1} \sum_{r=p-1}^{q-1} e^{2\pi i r b/L} \sum_{b'=0}^{L-1} e^{-2\pi i r b'/L} Z_L f(a,b'), \quad 0 \le a < M, \quad 0 \le b < L.$$

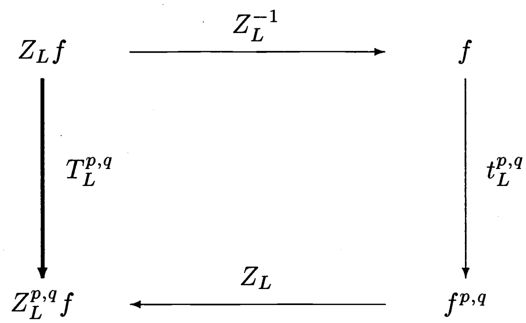

Figure 1: Diagram of the Zak space translation of the truncation operation. Bold arrow marks the reduced complexity operation of the new approach.

Proof: The result follows directly from application of (5) to (6).

In particular, for $p = q = 1$, we have $Z_L^{1,1} f(a,b) = T_L^{1,1} Z_L f(a,b) = L^{-1} \sum_{b'=0}^{L-1} Z_L f(a,b')$, and for $p = 1$ and $q = L$, we have $Z_L^{1,L} f(a,b) = T_L^{1,L} Z_L f(a,b) = Z_L f(a,b)$.

Theorem 1 together with theorem 2 leads to a new efficient formulation of the GP algorithm, where all constituent operations of the algorithm, i.e. signal truncation, bandlimitness constraint and domain translation are performed in a singular space of the FZT. In each iteration of the Zak space description of the GP algorithm we have to replace $Z_L f$ by $Z_L^{p,q} f$ as f varies over the iteration. The direct approach which is equivalent to the Fourier transform approach computes the inverse Zak transform of $Z_L f$, truncates and computes the Zak transform of the truncation. The theorem replaces these steps by the computation of $T_L^{p,q}$ on $Z_L f$ (Figure 1).

Table 1 summarizes computational complexity of the basic Zak space operations: the Zak transform, the Fourier transform, and the truncation, where the action of F_N is assumed to require $N log N$ additions and multiplications. The actions of $Z_M F_N Z_L^{-1}$ and $T_L^{p,q}$ are implemented using theorem 1 and theorem 2.

In the next section we give the Zak space formulation of the Gerchberg-Papoulis algorithm and demonstrate the reduction in computational complexity achieved by the new algorithm.

3. GERCHBERG-PAPOULIS-ZAK ALGORITHM

Consider a signal $f \in \mathcal{C}^N$, which we write as

$$f = \begin{bmatrix} f_0 \\ \vdots \\ f_{L-1} \end{bmatrix}, \quad f_l \in \mathcal{C}^M, \quad 0 \leq l < L. \tag{7}$$

Suppose

$$f_{p-1}, \ldots, f_{q-1}$$

are the known segments of f, and set

$$f^0 = t_L^{p,q} f.$$

f^0 depends solely on the known part of f and will be the initial guess in the iteration. Write $\mathbf{f} = F_N f$ as

$$\mathbf{f} = \begin{bmatrix} \mathbf{f}_0 \\ \vdots \\ \mathbf{f}_{M-1} \end{bmatrix}, \quad \mathbf{f}_m \in \mathcal{C}^L, \quad 0 \leq m < M. \tag{8}$$

Suppose

$$\mathbf{f}_{p'-1}, \ldots, \mathbf{f}_{q'-1}$$

are the nonzero segments of $\mathbf{f}$. f can then be recovered by the following procedure.

<u>Algorithm ($p,\ p' \geq 1$).</u>

Precomputation:

- compute $Z_L f^0$.

Iteration:

1. compute

$$Z_M \mathbf{f}^k = Z_M F_N f^k = (Z_M F_N Z_L^{-1})(Z_L f^k).$$

   This step requires N multiplications.

2. compute

$$Z_M \mathbf{g}^k = T_M^{p',q'} Z_M \mathbf{f}^k$$

   imposing the frequency domain constraint. This step requires $N(q'-p') + (N-L)(q'-p'+1)$ additions and $2N(q'-p'+1)$ multiplications for $p' > 1$ and $2N(q'-1)$ multiplications for $p' = 1$.

3. compute

$$Z_L g^k = (Z_L F_N^{-1} Z_M^{-1})(Z_M \mathbf{g}^k).$$

   This step requires N multiplications.

4. compute

$$Z_L h^k = (I - T_L^{p,q}) Z_L g^k,$$

   the Zak space truncation of the unknown samples. This stage requires $N(q-p) + (N-M)(q-p+1)$ additions and $2N(q-p+1)$ multiplications for $p > 1$ and $2N(q-1)$ multiplications for $p = 1$.

5. compute

$$Z_L f^{k+1} = Z_L f^0 + Z_L h^k$$

   imposing the known samples constraint. This step requires N additions.

Postcomputation:

- compute $Z_L^{-1}(Z_L f^{k+1})$.

The complete iteration can be expressed as

$$Z_L f^{k+1} = Z_L f^0 + (I - T_L^{p,q})(Z_L F_N^{-1} Z_M^{-1}) T_M^{p',q'} (Z_M F_N Z_L^{-1})(Z_L f^k)$$

and requires

$$2N(q' + q - p' - p + 1) - L(q' - p' + 1) - M(q - p + 1)$$

additions and

$$2N(q' + q - p' - p + 3)$$

multiplications for $p, p' > 1$ and

$$2N(q' + q - 1)$$

multiplications for $p = p' = 1$.

The diagram in Figure 2 compares the direct Zak space translation of the Gerchberg-Papoulis algorithm and the present approach. In the standard, or the direct Zak space translation approach

$$2N(logL + logM)$$

multiplications are required to implement the two truncations as compared with

$$2N(q' + q - p' - p + 2)$$

in the present approach. The present approach has the advantage whenever

$$q' + q - p' - p + 2 < logL + logM.$$

If $p = 1$ and $p' = 1$ then

$$2N(q' + q - 2)$$

multiplications are required to implement the two truncations, and the present approach has the advantage whenever

$$q' + q - 2 < logL + logM. \tag{9}$$

If $q' = q = 1$ then no complex multiplications are required by the present approach.

Inequality (7) determines the range of applications of the new approach in terms of small values of q and q'. In fact, the actual range of applications is much broader. It includes cases where either q, or $L - q$, and q', or $M - q'$, are small, i.e. when

$$min\{q, L - q\} + min\{q', M - q'\} - 2 < logL + logM. \tag{10}$$

Example 1

Take $p = 1,\ q = 62,\ p' = q' = 1,\ L = M = 64$. Steps 1-3 and 5 are implemented directly as described by the algorithm. The Zak space truncation of step 4 can be implemented either as $(I - T_{64}^{1,62})$ or as $T_{64}^{63,64}$. Since the factor $q - p + 1$ associated with the second form is much smaller, the second form is preferable. The iteration requires no multiplications for truncation of step 2 and $4N$ multiplications for truncation of step 4. The total number of multiplications required by the iteration is $6N$, which is half the number required by the direct approach.

When comparing the direct and the Zak space implementations of the Gerchberg-Papoulis algorithm, one has to take into account possible efficiency improvements of the iteration, that can be accomplished in the original, alternating space setting. These improvements can be made by taking advantage of the fact that if $q - p$ or $q' - p'$ is significantly smaller then M or L, then certain multiplications in computation of the forward or inverse DFT can be omitted. Several techniques, generally referred to as FFT pruning methods, have been developed in the past to take advantage of data sparsity to reduce complexity of the computation [8,11,12]. Here, we will consider pruning of only the forward Fourier transform, since the computation of the forward transform limits the overall efficiency of the pruned GP algorithm.

Table 2 provides a comparison between the multiplicative complexity of the pruning and Zak space methods for a N-input points/$q'L$-output points DFT. Here, we take $p' = 1$ and consider the Zak transform on a square $M \times L = \sqrt{N} \times \sqrt{N}$ lattice. The first technique requires $Nlog(q'L)$ operations (which is $log(q'L)/logN$ less then the N-input/N-output DFT), the second method requires $N(2q'-1)$ operations for small q' and $N(2M-2q'+1)$ operations for large q' (steps 1 and 2 of the algorithm).

The Zak transform method compares favorably with the pruning method for $L > 2^{2q'-1}/q'$. The superiority of the Zak transform method becomes even more pronounced if finer resolution is required of the DFT output, i.e. when q' grows while the ratio q'/M remains constant. As noted before, the Zak transform technique is effective for both small and large q', while the multiplicative complexity of the pruning method approaches $NlogN$, as q' becomes large.

An additional benefit of the Zak space method is in applications, where data samples are known on adjacent intervals. This occurs, for example, in data storage and transmission, where unknown samples often exhibit a scattered pattern [14]. Since truncation involves only pointwise multiplications, the computational complexity of the iteration depends only on the number of points to be computed and not on their location. Regardless of how the support of the time and frequency signals is defined, the steps 2 and 4 of the algorithm require $2L$ and $2M$ multiplications for every data point not restricted to the basic intervals $[0, M)$, and $[0, L)$, respectively. This is in contrast to the pruning methods, which require consecutive samples to achieve reduction in computational complexity [12].

4. SUMMARY

We have considered a new formulation of the Gerchberg-Papoulis algorithm, obtained by means of the finite Zak transform. It was shown, that the Zak space signal extrapolation is more efficient then the direct GP method by a factor of $logN$, and more efficient then the direct GP method combined with pruning by a factor of $log(q'L)$, when $q = q' = 1$. A significant but smaller reduction in the multiplicative complexity is achieved when q and q', or $L - q$ and $M - q'$ are small. In contrast to pruning, the new method is applicable in cases when the number of non-zero Fourier transform coefficients is large in comparison with N, and when the signal is bandlimited to a union of non-contiguous intervals. Since the Zak space method involves only pointwise multiplications, the GPZ algorithm is easier to code and more suitable for implementation in a parallel processing environment then the GP algorithm. Since the multidimensional finite Zak transform is well established [16], the Gerchberg-Papoulis-Zak algorithm can be easily extended to a higher-dimensional case, where the computational advantage is even more pronounced.

The GPZ algorithm proposed in this work assumes a complex valued signal, while many extrapolation problems involve real valued signals. In future work, we will introduce a new, real valued time-frequency transform, which we will call the real Zak transform (RZT). We will show that the RZT leads to a real Zak space formulation of the GP algorithm that is formaly similar to the approach presented here, but acts directly on a real signal via the real Zak space relation between a signal and its Hartley transform, leading in effect to approximately a fourfold reduction in the computational complexity of the GPZ approach.

BIBLIOGRAPHY

[1] M. An, A. K. Brodzik, I. Gertner and R. Tolimieri, "Weyl-Heisenberg Systems and the Finite Zak Transform", in *Signal and Image Representation in Combined Spaces*, eds Y. Zeevi and R. Coifman, pp 3-22, Academic Press, San Diego, 1998.

[2] L. Auslander, I. Gertner and R. Tolimieri, "Finite Zak Transforms and the Finite Fourier Transforms", *IMA on Radar and Sonar*, Part II, Vol. 39, pp 21-36, Springer-Verlag, New York, 1991.

[3] M. J. Baastians, "A Sampling Theorem for Complex Spectrogram and Gabor Expansion of a Signal into Gaussian Elementary Signals", *Opt. Eng.*, Vol. 20, No. 4, pp 594-598, 1981.

[4] A. K. Brodzik, "The Design of Discrete Gabor Expansion Algorithms and An Efficient Realization of the Gerchberg-Papoulis Algorithm", *Ph.D. Dissertation*, City University of New York, 1995.

[5] A. K. Brodzik and R. Tolimieri, "The Computation of Weyl-Heisenberg Coefficients for Critically Sampled and Oversampled Signals", *Proceedings IEEE-SP Symposium on Time-Frequency and Time-Scale Analysis*, Philadelphia, pp 272-275, 1994.

[6] R.W. Gerchberg, "Super-resolution through error energy reduction", *Opt. Acta.*, Vol. 21, No. 9, pp 709-720, 1974.

[7] A.J.E.M. Janssen, "The Zak Transform: A Signal Transform for Sampled Time-Continuous Signals", *Philips J. Res.*, Vol. 43, pp 23-69, 1988.

[8] J. D. Markel, "FFT Pruning", *IEEE Trans. ASSP*, Vol. 19, No. 4, pp 305-311, December 1971.

[9] A. Papoulis, "A New Algorithm in Spectral Analysis and Band-Limited Extrapolation", *IEEE Trans. Circ. Sys.*, Vol. CAS-22, No. 9, pp 735-742, 1975.

[10] N. Polyak and W. A. Pearlman, "Filters and Filter Banks for Periodic Signals, The Discrete Zak Transform and The Wavelet Transform Decomposition", *IEEE Trans. SP*, Vol. 46, No. 4, pp 857-873, April 1998.

[11] D. P. Skinner, "Pruning the Decimation In-Time FFT Algorithm", *IEEE Trans. ASSP*, Vol. ASSP-24, No. 2, pp 193-194, April 1976.

[12] T. V. Sreenivas and P. V. Rao, "FFT Algorithm for Both Input and Output Pruning", *IEEE Trans. ASSP*, Vol. 27, No. 3, pp 291-292, June 1979.

[13] R. Tolimieri and M. An, "Time-Frequency Representations", Birkhauser, Boston, 1998.

[14] R. Veldhuis, "Restoration of Lost Samples in Digital Signals", Prentice Hall, New York, 1990.

[15] J. Zak, "Finite Translations in Solid State Physics", *Phys. Rev. Lett.*, Vol. 19, pp 1385-1397, 1967.

[16] Y. Zeevi and I. Gertner, "The Finite Zak Transform: An Efficient Tool for Image Representation and Analysis", *J. Visual Comm. and Image Representation*, Vol. 3, pp 13-23, 1992.

operation	additions	multiplications
Z_L or Z_L^{-1}	$NlogL$	$NlogL$
$Z_M F_N Z_L^{-1}$ or $Z_L F_N^{-1} Z_M^{-1}$	0	N
$T_L^{p,q}$, $p = 1$	$N(q-1) + (N-M)q$	$2N(q-1)$
$T_L^{p,q}$, $p > 1$	$N(q-p) + (N-M)(q-p+1)$	$2N(q-p+1)$

Table 1: Computational complexity of the basic Zak space operations.

			$N = 1024$				
$q'L$	32	64	128	...	896	960	992
pruning	5120	6144	7168	...	10240	10240	10240
Zak	1024	3072	7168	...	9216	5120	3072
			$N = 4096$				
$q'L$	64	128	256	...	3840	3968	4032
pruning	24576	28672	32768	...	49152	49152	49152
Zak	4096	12288	28672	...	36864	20480	12288
			$N = 16384$				
$q'L$	128	256	512	...	15872	16128	16256
pruning	114688	131072	147456	...	229376	229376	229376
Zak	16384	49152	114688	...	147456	81920	49152

Table 2: Comparison between the multiplicative complexity of the pruning and Zak transform methods.

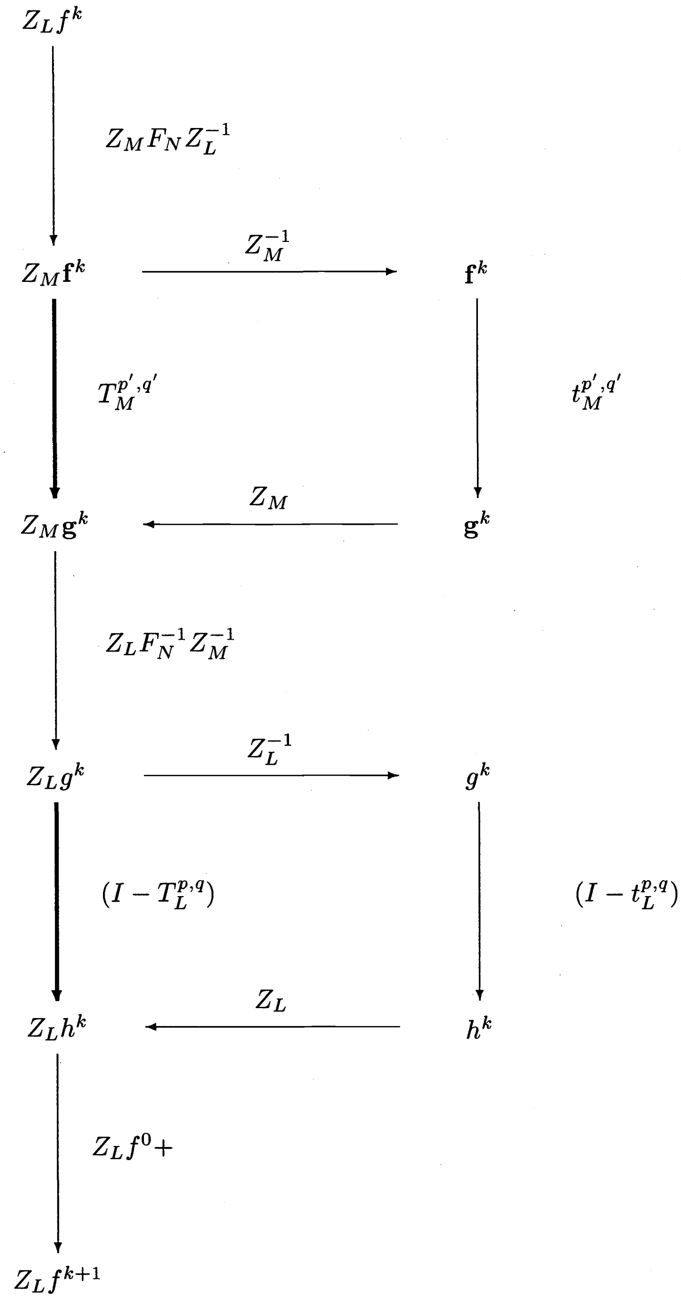

Figure 2: Diagram of the Zak space translation of the GP algorithm. Bold arrows mark reduced complexity operations of the new approach.