PERFORMANCE AND COMPUTATION RANKING OF FAST UNITARY TRANSFORMS IN APPLICATIONS*

V. Ralph Algazi* and Bernard J. Fino**

*Signal and Image Processing Laboratory, Department of Electrical and Computer Engineering, University of California, Davis, CA 95616, USA
**TRT, 5 Avenue Reaumur, B.P. 21, 92350 Le Plessis Robinson, FRANCE

1. ABSTRACT AND INTRODUCTION

For several signal processing applications, the usefulness of Fast Unitary Transforms (FUT) is now well recognized [1-7]. For signal representation, filtering and encoding, it is well known that the Karhunen-Loeve (KL) Transform, based on signal statistics, is optimum in various senses, but the KL Transform is slow. Suboptimum FUT's allow a trade-off between performance and speed.

In this paper, we compare and rank the KL, Fourier, Walsh-Hadamard, Haar, Discrete Cosine, Slant Walsh Hadamard and Slant Haar Transforms by their performance in applications and by the number of elementary operations they require.

In encoding and filtering, recursive techniques are widely used and are generally fast. By considering both performance and computations we are able to compare directly recursive and transform algorithms. The comparison brings to light a performance versus computation bound for the two classes of processing techniques.

2. A UNIFIED VIEW OF FAST UNITARY TRANSFORMS [FUT]

We have developed and reported [8] a unified approach to the generation of most known FUT and of new transforms. We review briefly the results presented in [8].

2.1 Recursive Generation of Fast Unitary Transforms

Note that most FUT have been proposed independently by different methods and are often analysed after the fact. In the approach presented in [8], new transforms can be recursively generated and the number of computations predicted. The approach is based on two synthesis rules.

i) FUT multiplication: If U,V are FUT with O_1, O_2 operations, UV is an FUT with O_1+O_2 operations.

ii) The generalized Kronecker Product of two sets of matrices $\{A\}$ of m unitary matrices $[A^i]$ $(i=0,\ldots,m-1)$ of order n and $\{B\}$ of n unitary matrices $[B^k]$ $(k=0,\ldots,n-1)$ of order m is defined as the matric [C] of order nm,

$$[C] \triangleq \{A\} \otimes \{B\}\ ,\ Cij = A^{w}_{uu'}\ B^{u'}_{ww'}$$

with $i = um + w$ and $u, u' = 0,\ldots,n-1$

$j = u'm + w'$ $\quad w, w' = 0,\ldots,m-1$

We have shown that [C] is unitary and can be factorized into

$$[C] = [P^t][\mathrm{Diag}\{A\}][P][\mathrm{Diag}\{B\}] \qquad (1)$$

where $P_{ki} = \delta_{vw}\delta_{zu}$, is a perfect shuffle with $k = vn+z$, $i = um+w$ and δ the Kronecker delta and

$$[\mathrm{Diag}\{A\}] = \begin{bmatrix} [A^0] & & 0 \\ & [A^1] & \\ 0 & & [A^{m-1}] \end{bmatrix}$$

When the matrices [A] (and [B]) are identical we have the usual Kronecker product of matrices. If the matrices $[A^i]$ and $[B^k]$ require respectively p^i and p^k elementary operations, transformation by the generalized Kronecker product [C] will require P such operations with

$$P = \sum_i p^i + \sum_k p^k \qquad (2)$$

By the combined use of these two matrix operations, we define recursively large classes of FUT. The Fast Fourier Transform with Cooley-Tukey algorithm is recursively defined at the order 2^n from the matrix of order 2^{n-1}

$$[F_{2^n}] = [\{F_2^0][F_2^1]\ldots[F_2^{(2^{n-1})-1}]\}$$

$$\otimes \{[F_{2^{n-1}}][F_{2^{n-1}}]\}][P^t] \qquad (3)$$

with

$$[F_2^k] = \frac{1}{\sqrt{2}} \begin{bmatrix} 1 & \exp(-2\pi jk/2^n) \\ 1 & -\exp(-2\pi jk/2^n) \end{bmatrix}$$

For the Fourier, Walsh-Hadamard and Haar, we have found recursive definitions with relations similar to (3). We generalize the W-H and Fourier transforms to a large family of FUT's given by

*This research was supported by the National Science Foundation under grant ENG-78-11521

CH 1746-7/82/0000 · 0032 $ 00.75 © 1982 IEEE

$$[GT_{2^n}]=[\{[F_2(\theta_0),..,F_2(\theta_{2^{n-1}})\}\otimes[GT_{2^{n-1}}]][P^t] \quad (4)$$

where $[F_2(\theta)] = \frac{1}{\sqrt{2}}\begin{bmatrix} 1 & \exp(-j\theta) \\ 1 & -\exp(-j\theta) \end{bmatrix}$

This family includes the W-H and Fourier transforms for the appropriate choices of the parameters $\theta_0 \cdots \theta_{2^{n-1}-1}$. Generalizations to families of FUT between the Walsh-Hadamard and Haar transform are also possible. We make use of relation (2) for the number of operations, and recursive definition (3) or (4) lead to a recursive expression for the number of operations needed to transform a vector by matrix $[GT_{2^n}]$.

A class of slant transforms can be obtained by modifying a transform with fast algorithm to include a prescribed set of vectors. A noteworthy new transform thus generated is the Slant Haar transform [9] which has the advantages of basic vectors of different length, just like the Haar Transform, and includes some slant vectors useful in the representation of slowly varying signals. The slant Haar Transform is also extremely fast.

2.2 Number of Operations

We consider the most common transforms and on the Slant Haar Transform. The transforms considered here are listed in Table 1.

TRANSFORM	ADDS	MULTIPL.	ADD&MULPL.
Haar (HA)	2-2/M	0	2-2/M
Slant Haar (SH)	4-6/M	1/4-1/M	4.25/7/M
Walsh-Hadamard (WH)	N	0	N
Slant Walsh Hadam(SW)	$N+\frac{1}{2}$-2/M	1/4-1/M	N+3/4-3/M
Discrete Fourier (FR)	$\frac{3}{2}(N-1)+\frac{2}{M}$	$\frac{1}{2}(N-3)+\frac{2}{M}$	2N-3+4/M
Discrete Cosine (DC)	$3(N-1)+\frac{4}{M}$	$(N-3)+\frac{4}{M}$	4N-6+8/M

Table 1. Number of Computations per Points BLOCK $M = 2^N$

We list the number of additions, and multiplications and their sum for use in applications. Note that the implementation of FUT requires data transfers, shifts or normalizations and that the speed of execution depends on the time of execution of each operation and of the architecture of the processor. We have found that on a PDP11 microcomputer using a floating point processor, execution time is approximately the same for additions and multiplications. Thus the add & multiply column gives an indication of the relative execution time of FUT algorithms on minicomputers. These results are confirmed experimentally. As an example, for M=128, N=7, we find that FUT's range 11 to 1 in number of operations. Note that FUT's have a number of operations which increases linearly with $N=\log_2 M$ except for the Haar and Slant Haar which have limits of 2 and 4.25 operations per point respectively. We shall also use, in applications, the KL Transform, for which we assume that the number of operations per point is 2M.

3. APPLICATIONS OF FUT

We now consider the applications of FUT's in each case we evaluate, for known signal statistics, the performance of the KL Transform and of FUT's. If appropriate, we also evaluate the performance of recursive algorithms. We assume that signals are stationary Gaussian sequences with correlation functions, $R_{xx}(i,j)=e^{-\alpha|i-j|}$ for a Gauss-Markov process, and $R_{xx}(i-j)=\frac{\sin\alpha(i-j)}{\alpha(i-j)}$, for an ideal low-pass process.

3.1 Signal Representation and Dimensionality Reduction

In many applications, the data is highly redundant and has too many dimensions for efficient processing. For example, in pattern recognition, before the classification stage, feature extraction will reduce dimensionality [10]. Unitary transformations can be used to provide efficient representations of signals. Let $\underline{X}$ be a signal vector if $\underline{Y} = T\underline{X}$, where T is a unitary transform, then the transform vector $\underline{Y}$ gives an exact representation of $\underline{X}$. Assume that some transform coefficients Y_k are modified or discarded to yield the approximate representation $\hat{\underline{Y}}$. The mean square error (m.s.e.) in the representation of $\underline{X}$ can be evaluated as

$$MSE(\underline{X}) = ||\hat{\underline{X}}-\underline{X}||^2 = ||\hat{\underline{Y}}-\underline{Y}||^2$$

with $\underline{X} = T^*\hat{\underline{Y}}$

where T* denotes the conjugate transpose of T. If we discard some coefficients Y_k, then the m.s.e. can be expressed in terms of T and of the covariance matrix R_{xx}, by

$$MSE(X) = \sum_{k\in \text{DISCARDED COEFFICIENTS}} [T\, R_{xx}\, T^*]_k \quad (5)$$

The KL Transform is optimal in resulting in the least m.s.e. for a fixed number of discarded coefficients [10] and also requires the least number of coefficients in the representation of Gaussian sequences for a fixed mean square error [11]. As an example, we consider the representation of a Gauss-Markov process with α=0.05 and we retain 1/4 of the coefficients. The performance is measured by the m.s.e. and referred to the m.s.e. of the KL Transform with M=128. The increase m.s.e., or loss, can be measured in dB. For each value of block size M, we obtain, from Table 1, the number of computations and the m.s.e. from (5). The block size is thus an implicit parameter in the graphs of performance versus number of computations. Results are shown in Figure 1.

3.2 Scalar Filtering of Signals [2,4]

Consider a noisy signal $\underline{S}=\underline{X}+\underline{Y}$ where $\underline{Y}$ is

uncorrelated noise.

If $\hat{\underline{X}}=H\underline{S}$ is the filtered signal, it is well known [2] that the filter matrix which minimizes $E[||\hat{\underline{X}}-\underline{X}||^2]$ is given by

$$H_{OPT} = R_{xx}[R_{xx}+R_{yy}]^{-1}$$

with

$$MSE(X) = E[||\hat{\underline{X}}-\underline{X}||^2] = Tr[R_{xx}(R_{xx}+R_{yy})^{-1}R_{xx}]$$

where Tr[ ] denotes the trace of the matrix. If a Unitary Transform T is applied before filtering then the optimal filter becomes $H'_{OPT}=THT^*$. If we constrain H to be a diagonal matrix H_D, it can be shown [4] that

$$H_{D,OPT} = Diag\frac{[TR_{xx}T^*]_{ii}}{[T(R_{xx}+R_{yy})T^*]_{ii}} \triangleq Diag[D_{ii}]$$

with

$$MSE(X) = 1 - \frac{1}{M}\sum_i D_{ii}[TR_{xx}T^*]_{ii} \qquad (6)$$

such a scheme is optimum if T is the KL Transform. Thus, each FUT results in a suboptimum MSE given by (6). As an example, we consider a Gauss-Markov process $\underline{X}$ and additive white noise $\underline{Y}$, and evaluate the loss in SNR, resulting from the use of each FUT, versus the number of computations. The reference is the KL Transform with M=128. We also show, on Figure 2, the performance of a first order recursive filter, denoted RC.

3.3 Signal Encoding [2,3,14,15]

Signal encoding by FUT's consist of three successive operations: the direct transform of a block of signal samples, the selection and quantization of some transform coefficients and the efficient binary encoding of the quantized transform coefficients. The decoder performs the inverse transform of the quantized coefficients. A transform encoding technique is thus determined by the choice of transform and block size, and by the method of selection of and quantization of the transform coefficients.

We assume here that the signal is Gaussian, that transform coefficients are encoded independently and that the Gaussian random variable X with variance σ^2 can be encoded with a m.s.e. distortion μ using on the average $r = \frac{1}{2}\log_2[\sigma^2/\mu]$ bits. This rate-distortion function $r(\mu)$ is the least number of bits needed to represent μ and can only be achieved approximately by uniform quantization of x and entropy coding [12].

For stationary Gaussian signals, the optimal encoder uses the KL Transform, quantizes all transform coefficients with the same mean square error and results in global rate distortion equations given parametrically by

$$R(\mu) = \frac{1}{2M}\sum_{\sigma_i^2>\mu} \log_2 \frac{\sigma_i^2}{\mu} \text{ bits}$$

$$D(\mu) = \frac{1}{M}\left(\sum_{\sigma_i^2>\mu} \mu + \sum_{\sigma_i^2\le\mu} \sigma_i^2\right) = E[||\hat{X}-X||^2] \qquad (8)$$

where $\{\sigma_i^2\}$ are the variances of the transform domain coefficients. For FUT's, we use the same encoding rules, but the variances will depend on the specific transform used.

As an example, we consider the encoding of a low pass signal with α=0.05 and a distortion D=.005. We show, in Figure 3, the bit rate versus the number of computations. We show on the same figure the performance of a differential pulse code modulation (DPCM) encoder using a uniform quantizer and linear prediction. The equations for DPCM performance can be found in [13].

The performance versus computation curves for all algorithms are bounded by an empirical curve shown on Figure 3 and denoted the performance computation bound. Note that a gap exists between DPCD and other fast FUT's (2-5 oper/point) and the performance of the discrete cosine transform. Thus it may be of interest to develop new transforms which would cover that gap.

4. DISCUSSION AND CONCLUSIONS

The point of view presented here provides some insight into the potential usefulness of various transform techniques in applications. Specific conclusions can be drawn.

i) Transform techniques which seem close to each other in performance are quite different computationally and thus can be ranked on the basis of performance versus computations.

ii) Large block FUT's and thus the asymptotic computational complexity of transforms have no significant merit in signal processing applications.

iii) In signal filtering and encoding a performance-computation bound exists for both recursive and transform techniques. Some fast transforms perform very well for very few operations per output point.

Note that this theoretical work has not considered adaptive techniques and that the mean-square error is often not an appropriate distortion measure in applications. For adaptive algorithms, short block transforms which can follow changing signal characteristics seem of prime interest. Research is being carried out by one of the authors in the application of the Slant Haar Transform to adaptive filtering and encoding.

(1) H.F. Harmuth, Transmission of Information by Orthogonal Functions, Springer, 1972.
(2) N. Ahmed and K.R. Rao, Orthogonal Transforms for Digital Signal Processing, Springer,1975.
(3) P.A. Wintz, "Transform picture coding," Proc. IEEE, 60, pp. 809-820:1972.
(4) W.K. Pratt, "Generalized Wiener filtering computation techniques," IEEE Trans., C-21, pp. 636-641:1972.
(5) H. Enomoto and K. Shibata, "Orthogonal transform coding system for television signals," Proc. 1971 Symp. Walsh Functions, pp. 11-17.
(6) W.K. Pratt, L.R. Welch and W.H. Chen, "Slant transform for image coding," Proc. 1972 Symp. Walsh Functions, pp. 229-234.
(7) N. Ahmed, T. Natarajan and K.R. Rao, "Discrete Cosine Transform," IEEE Trans., C-23, pp. .90-93:Jan. 1974.
(8) B.J. Fino and V.R. Algazi, "A Unified Treatment of Discrete Fast Unitary Transforms," SIAM J. COMPUT., Vol. 6, No. 4, 1974.
(9) B.J. Fino and V. R. Algazi, "Slant Haar transform," Proc. IEEE, 62, 5, 1974.
(10) H. C. Andrews, An Introduction to Mathematical Techniques in Pattern Recognition, Wiley, 1972.
(11) V. R. Algazi and D.J. Sakrison, "On the Optimality of the Karhunen Loeve Transformation," IEEE Trans., IT-15, 1969.
(12) R.G. Gallagher, Information Theory and Reliable Communications, Wiley, 1968.
(13) V. R. Algazi and J.T. DeWitte, "Theoretical Performance of Entropy Encoded DPCM," IEEE Trans. Comm., (in press).
(14) S.J. Campanella and G.S. Robinson, "A comparison of orthogonal transformations for digital speech processing," IEEE Trans., COM-19, pp. 1045-1050:1971.
(15) R. Zelinski and P. Noll, "Adaptive Transf. Coding of Speech Signals," IEEE Trans., ASSP-25, No. 4, 1977.

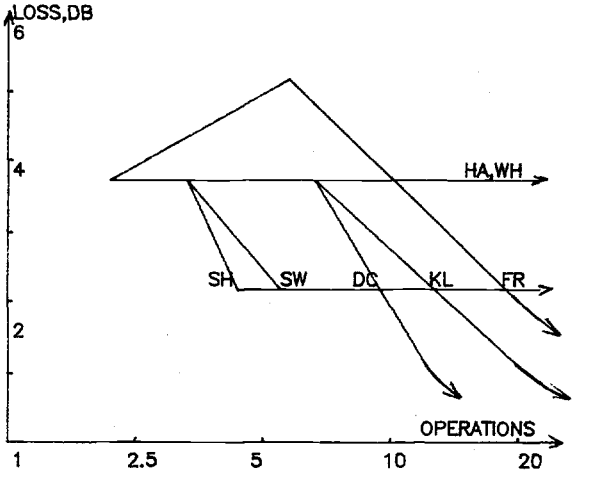

FIG.1:REPRESENTATION,GAUSS–MARKOV SIGNAL

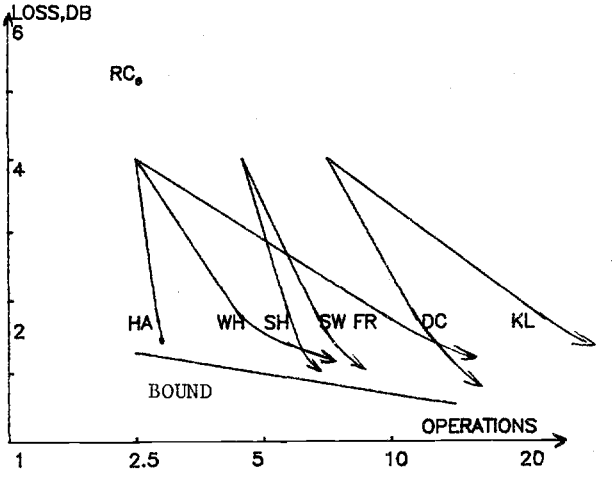

FIG.2:FILTERING,GAUSS–MARKOV SIGNAL(SNR=1)

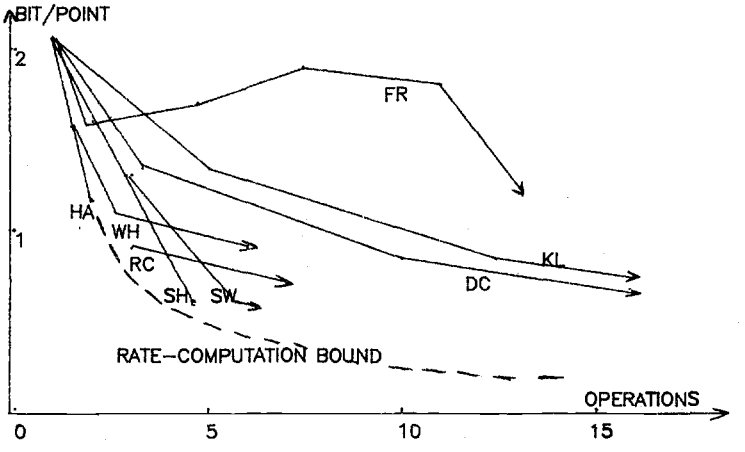

FIG.3 ENCODING:LOW PASS SIGNAL,DISTORTION=0.5%