COPERM: TRANSFORM-DOMAIN ENERGY COMPACTION BY OPTIMAL PERMUTATION

Nicholas D. Sidiropoulos

Dept. of Electrical Engineering
University of Virginia
Charlottesville VA 22903 U.S.A.
nikos@virginia.edu

*Marios S. Pattichis, Alan C. Bovik, John W. Havlicek**

Dept. of Electrical & Computer Engineering
University of Texas at Austin
Austin, TX 78712 U.S.A.
<marios,bovik>@ece.utexas.edu

ABSTRACT

COPERM is a novel paradigm for energy compaction and signal compression, whose foundation is a simple but powerful idea: any signal can be transformed to resemble a more desirable signal from a class of "target" signals, by means of a suitable permutation of its samples. The approach is well-suited for transform domain energy compaction prior to transform-domain compression of persistent broadband signals. The associated optimal *permutation precoders* are surprisingly simple, and the permutation precoding overhead can be made modest - resulting in improved overall rate-distortion performance.

1. INTRODUCTION

Energy compaction is an important first step in almost all compression techniques [1]. Prediction and/or transformation to some other more suitable domain are the two prevailing techniques for energy compaction. Both of these energy-compaction techniques are not very efficient in compacting the energy of persistent broadband signals. These signals occupy a wide bandwidth in the frequency domain; and exhibit fast and persistent variation in the time domain. For such signals, wavelet analysis is an attractive option. In this paper, we introduce another attractive option, which is, in fact, closely related to optimal signal-adaptive AM-FM signal analysis and synthesis. In particular, we propose a novel technique for energy compaction of broadband signals via optimal permutation. This permutation is matched to a particular transform-domain codec, in the sense that its goal is to transform the input into a signal that is customized to fit the strengths of the transform-domain codec.

This research was supported in part by the Army Research Office under grant DAAH 04951O494 and by the Air Force Office of Scientific Research, Air Force Systems Command, USAF, under grant number F49620-93-1-0307.

* John W. Havlicek (havlicek@cs.utexas.edu) is with the Dept. of Computer Science at the University of Texas at Austin.

We present some basic theoretical results that lead to fast computational algorithms, and illustrate key concepts by using 1-D examples, as well as designing a hybrid image codec that uses JPEG in conjunction with COPERM pre - processing to improve on the rate-distortion performance of JPEG in the high-quality operating region. This region is important in critical compression applications (e.g., in biomedicine).

The idea of using permutations for (stand-alone) source coding has been investigated in the mid 60's to early 80's by Dunn [2], and Berger *et al.* [3], motivated from a channel coding proposal of Slepian [4]. Alternative raster scan-orders (e.g., Hilbert curve) in lossless image coding can be viewed as *local prediction-error energy*-compacting permutations [5]. These are the closest pieces of prior work. References [6, 7] report on some other interesting signal processing applications of permutations.

2. MAIN IDEA

We introduce the main idea using the Discrete Fourier Transform (DFT). The DFT is closely related to the Discrete Cosine Transform (DCT), which is heavily used as a pre - processing energy - compaction block in many transform-domain codecs, like JPEG [1]. These codecs also incorporate quantization and entropy encoding blocks. We temporarily leave these blocks out of consideration (they too can be accounted for), and focus on energy-compaction.

What is the best possible DFT input signal from an energy compaction viewpoint? Clearly, it is[1] *one* of the given DFT basis signals. Many real-life signals are smooth, thus having most of their energy in the low-pass-band of the spectrum: the DFT is quite effective in compacting the energy of such signals. Actually, it is quite effective in compacting the energy of any narrowband signal, regardless of it being low-pass or band-pass.

[1] Modulo scaling by a (potentially complex) number, of course.

0-7803-4428-6/98 $10.00

The flip-side of this is that the DFT is a poor choice for compacting the energy of broadband signals, e.g., textures, fingerprints, noise, fractals, or certain digital modulation signals. For such signals, wavelet analysis often is an attractive option. In this section, we introduce another attractive option. Suppose we are given such a broadband signal. Is it possible to somehow *transform* this signal in such a way that: (i) The transformed signal is as narrowband as possible, ideally one of the given DFT basis functions; (ii) The associated transformation is *invertible*, so that we may recover the original signal from the transformed signal; (iii) The transformation can be encoded at a moderate coding cost (some coding overhead is clearly necessary; the idea is to come up with an improved overall balance); and (iv) The forward and inverse transforms are easy to *construct*, and *apply*?

For the sake of simplicity, and without loss of generality, let us temporarily restrict our attention to so-called *constant modulus* signals. The reason for temporarily restricting ourselves to this class of signals is not technical but pedagogical: it allows us to make the first argument using the Discrete Fourier Transform, a very familiar tool. The argument carries through for real-valued signals, and certain complex signals as well.

Constant modulus signals are complex-valued signals whose magnitude remains constant over time, and only their angle changes: a constant modulus signal is any signal that can be written as $x(n) = Ae^{j\alpha(n)}$, where A is a constant, $\alpha(\cdot)$ is a real-valued function, and j is the square root of -1. Note that the *angle function* of $x(n)$ is not $\alpha(n)$, but rather $\alpha(n)$ modulo 2π.

Figure 1(a) depicts the angle function of a broadband constant-modulus signal; Figure 1(d) depicts the magnitude of the DFT of this signal. Figure 1(e) depicts the magnitude of the DFT of a suitable permutation of the given constant-modulus signal: notice that the permuted signal essentially consists of a single DFT harmonic (there exist a few more negligible but nonzero DFT coefficients which are not visible in this plot). Figure 1(b) depicts the angle function of the reconstructed signal, obtained by setting all negligible DFT coefficients of the permuted signal to zero, computing the inverse DFT, and then de-permuting using the inverse permutation. This effectively re-creates the angle function of the original constant-modulus signal. Figure 1(c) depicts the error between the angle functions of the original signal and the reconstructed signal (SNR is about 100dB). Of course, from a compression standpoint, one has to consider the cost of storing not only the compacted transform (one complex number in this case), but also the associated permutation. There exist $N!$ possible permutations of a length-N signal - meaning that $log(N!)$ bits are generally required to represent an arbitrary length-N signal permutation. This is slightly better than $Nlog(N)$, or $log(N)$ bits per signal

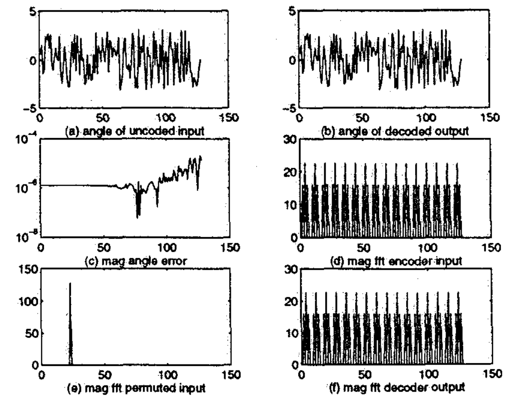

Figure 1: $x(n) = e^{j\frac{2\pi}{N}(1+3n-2n^2+12n^3-n^6)}$.

sample. Thus it appears that N should not be too big (actually: if the signal is digital, then $log(N)$ should be strictly less than the number of bits used to represent a signal sample in order to achieve any compression gain at all). Of course, if the signal is real-valued this is less of an issue. On the other hand, N should not be too small, for otherwise even the optimal matching is often not very accurate in absolute terms. It appears then that the choice of N exhibits an interesting (and unusual) trade-off. However, as we will soon see, there is an elegant way to circumvent the restriction of having to work with relatively small N, even for digital data.

How does one find such a *suitable* permutation?

Recall that the permutation should ideally transform the given signal to a DFT basis function. We cannot insist on exact transformation, for there exist signals which cannot be synthesized by permutation of a DFT basis function. It seems natural, then, to pose the following problem: Given a constant modulus signal, find a permutation that optimally matches the angle function of the permuted signal to the angle function of *some* DFT basis signal, i.e., best matches the angle function of the permuted signal to $(\frac{2\pi}{N}kn)mod(2\pi)$, for the best possible $k \in \{0, \cdots, N-1\}$, in a Least Squares (LS) sense.

Let us use the letters ϕ, θ, r to denote permutations of the integers between 0 and $N-1$. Let $\mathcal{G}$ be the group of all such permutations ($|\mathcal{G}| = N!$). In concise mathematical terms, the problem can be stated as follows: Given the angle function, $p(n)$, of a constant element modulus input signal, find $r^* \in \mathcal{G}$ to *minimize the following minimum* (double minimization):

$$min_{k\in\{0,1,\cdots,N-1\}} \sum_{n=0}^{N-1} |p(r(n)) - (\frac{2\pi}{N}kn) \; mod \; (2\pi)|^2 \quad (1)$$

2.1. Some Technical Results

Given a real-valued sequence $p(n)$ of length N, we will say that ϕ is a *sorting permutation* for $p(n)$ if the sequence $p(\phi(n))$ is sorted in increasing order (sorting permutations are not necessarily unique, because of the possible existence of *ties*: two or more elements may have exactly the same value).

Theorem 1 (Var. of a result of [4], [3]) *Given two real sequences of length N, $p(n)$ and $\alpha(n)$, consider the following problem:*

$$\textbf{minimize}: \sum_{n=0}^{N-1} |p(r(n)) - \alpha(n)|^2$$

$$\textbf{subject to}: r \in \mathcal{G}$$

Let ϕ, θ be sorting permutations for $p(n)$, $\alpha(n)$ respectively. Then an optimum r is given by $r(\theta(n)) = \phi(n)$, $\forall\, n$.

Sorting is, at worst, an $O(Nlog(N))$ operation, hence an optimum r can be found in $O(Nlog(N))$ operations. It is now clear how to solve the optimization problem in Equation (1): for fixed k, invoke the above Theorem to solve for the best r for the given k. This takes $O(Nlog(N))$ operations. Repeat for all N different k's and pick the best such r (for the best k) as the final answer. The overall process entails $O(N^2 log(N))$ operations.

There exists an interesting alternative to the above Theorem for the special case of finite-alphabet sequences:

Theorem 2 (Proof: [8]) *Consider two finite - alphabet sequences of length N, $p(n)$ and $\alpha(n)$. Assume that the alphabets are fixed and known in advance (this can always be assumed for the "target" sequence, $\alpha(n)$). Let h_p and h_α be the histograms of the two sequences over the respective alphabets; and $M = max(length(h_p), length(h_\alpha))$ (i.e., equal to the size of the largest alphabet). Then:*

$$minimum_{r\in\mathcal{G}} \sum_{n=0}^{N-1} |p(r(n)) - \alpha(n)|^2 = f(h_p, h_\alpha) = f^*$$

and f^* can be computed from h_p and h_α in $O(M)$ operations.

In light of this result, an alternative approach emerges: given a finite-alphabet input sequence and a number of target sequences, find the best target sequence by (i) computing the histogram of the input sequence in $O(N)$ operations (the corresponding computation for the target sequences need only be done *once* and *off-line*); and (ii) select the best target sequence by computing the associated f^* metrics, as above, at a complexity cost of $O(M)$ operations each. In the end, one only needs *one* sorting to find the optimum match. This being a finite-alphabet sorting operation, it follows that the overall runtime complexity of this process is $O(N + T \times M + N) = O(N + T \times M)$ operations, where T is the number of target sequences, and M is maximum alphabet size. This is important for M small relative to N.

A second example is shown in Figure 2(a)-(f), following the same format as Figure 1. As in the first example, just *one* (the strongest) DFT coefficient is sufficient to adequately represent the optimally permuted signal!

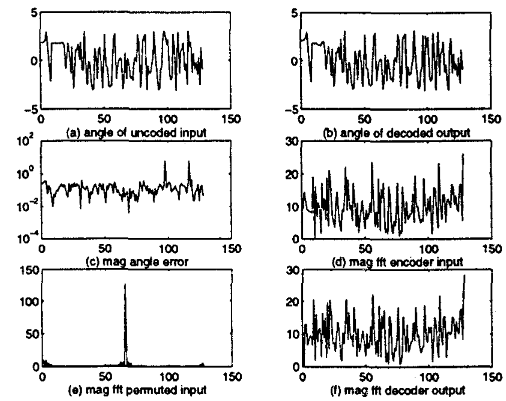

Figure 2: $x(n) = e^{j\frac{2\pi}{N}(\Gamma(n)+36)}$. $\Gamma(\cdot)$ is the gamma function.

2.2. System Design

One way to keep search and permutation coding overhead costs in check is to exploit the inherent *periodicity* of, e.g., DFT/DCT, basis functions in the time domain. Consider the DFT basis function for $k = N/2$. This function is a binary oscillation: $(-1)^n$. When sorted, it gives two "buckets": any sample of the original sequence may fall in *one* of the two buckets; exactly where it falls within a given bucket is irrelevant in terms of LS fit. Consequently, a single bit index per signal sample is sufficient to determine *an* optimal signal permutation for the given k. In general, one needs $log(N/k)$ bits per signal sample to represent an optimal permutation for $k = 1, 2, 4, \cdots, N/2$. Restricting the search to these frequencies also lowers complexity to $O(N(log(N))^2)$.

We have designed a coder that consists of an energy-compaction front-end using Theorems 1, 2 in conjunction with 1-D DCT for energy compaction, the above bucketing scheme for effectively reducing the permutation coding overhead, and JPEG for compressing the permuted image. The details can be found in [8]. The decision on what is the best k for a given image block (including whether or not to permute at all) is based on minimizing $cost(k) = log(N/k) + \lambda(1 - R(k))$, where λ is a trade-off parameter, and $R(k)$ is the ratio of the energy of the strongest DCT coefficient of the optimally permuted (for the given k) signal, divided by total signal energy.

Consider the $8bpp$, 512×512 fingerprint image, shown in Figure 3. Figure 4 shows the optimally permuted fingerprint image: this is input to the JPEG coder. Notice that only blocks which can clearly benefit from permutation are actually permuted; optimal permutation effectively transforms these blocks into almost - harmonic signals, which are particularly easy to compress via JPEG. For both schemes (i.e., with or without permutation precoding), different rate-distortion trade-offs were achieved by varying the quality parameter of JPEG. The hybrid codec loses its ground at relatively small rates; this is to be expected, due to the overhead associated with coding the permutation. However, at higher bit rates (high-quality operational region, beyond 3 bpp, which corresponds to quality factor ≥ 75 on the JPEG quality dial), improved energy compaction more than pays for the permutation coding overhead, and the hybrid codec wins uniformly over JPEG. For example, at a rate of 3.758 bpp the hybrid codec provides 35.44 dB's of PSNR, whereas at 3.744 bpp JPEG provides 34.82 dB's of PSNR.

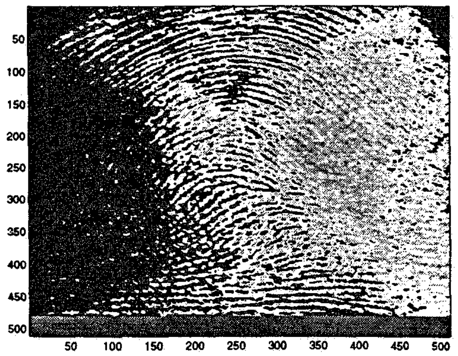

Figure 3: 8bpp fingerprint image.

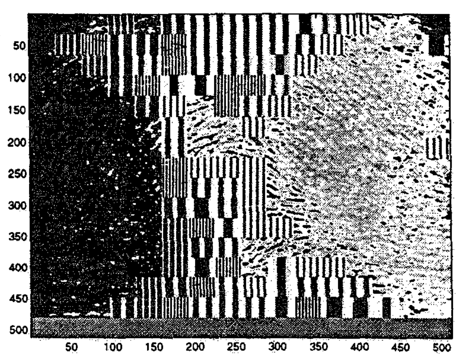

Figure 4: Optimally permuted fingerprint image.

3. CONCLUSIONS AND ON-GOING WORK

We have mentioned that the ideas presented herein have close ties to AM-FM signal analysis and synthesis. Indeed, our present work has been motivated from the pursuit of optimal signal-adaptive FM transforms [8]. There exist several promising extensions (e.g., in the direction of optimal AM-FM transforms), and these are the subject of on-going investigation.

COPERM decomposes the input signal in two components: a narrowband "unmodulated carrier-like" component, and a "noise-like" permutation component. Both components individually appear to be inconspicuous transmissions. Proper reconstruction requires both components, and is very sensitive with respect to the parameters of the unmodulated carrier - which are therefore natural candidates for a "key" in secure communications of compressed data.

4. REFERENCES

[1] V. Bhaskaran and K. Konstantinides. *Image and Video Compression Standards - Algorithms and Architectures.* Kluwer Academic Publishers, Boston, 1996.

[2] J.G. Dunn. *Coding for continuous sources and channels.* PhD thesis, Columbia University, New York, NY, 1965.

[3] T. Berger, F. Jelinek, and J.K. Wolf. Permutation Codes for Sources. *IEEE Trans. Information Theory,* IT-18:160–169, Jan. 1972.

[4] D. Slepian. Permutation Modulation. *Proc. IEEE,* 53:228–236, Mar. 1965.

[5] N.D. Memon and K. Sayood. Lossless Image Compression - A Comparative Study. In *Still Image Compression, Proc. SPIE 2418,* pages 8–20, 1995.

[6] K.E. Barner and G.R. Arce. Permutation filters: A class of nonlinear filters based on set permutations. *IEEE Trans. Signal Processing,* 42(4):782–798, 1994.

[7] Z. Arnavut. Applications of inversions to lossless image compression. *Optical Engineering,* 36(4):1028–1034, April 1997.

[8] N.D. Sidiropoulos, M.S. Pattichis, A.C. Bovik, and J.W. Havlicek. COPERM: Transform-domain energy compaction by optimal permutation. *IEEE Trans. on Signal Processing.* Submitted.