Frame permutation quantization ☆,☆☆

Ha Q. Nguyen [a], Vivek K. Goyal [a,*], Lav R. Varshney [a,b]

[a] *Department of Electrical Engineering and Computer Science and the Research Laboratory of Electronics, Massachusetts Institute of Technology, Cambridge, MA 02139, USA*
[b] *Laboratory for Information and Decision Systems, Massachusetts Institute of Technology, Cambridge, MA 02139, USA*

A B S T R A C T

Frame permutation quantization (FPQ) is a new vector quantization technique using finite frames. In FPQ, a vector is encoded using a permutation source code to quantize its frame expansion. This means that the encoding is a partial ordering of the frame expansion coefficients. Compared to ordinary permutation source coding, FPQ produces a greater number of possible quantization rates and a higher maximum rate. Various representations for the partitions induced by FPQ are presented, and reconstruction algorithms based on linear programming, quadratic programming, and recursive orthogonal projection are derived. Implementations of the linear and quadratic programming algorithms for uniform and Gaussian sources show performance improvements over entropy-constrained scalar quantization for certain combinations of vector dimension and coding rate. Monte Carlo evaluation of the recursive algorithm shows that mean-squared error (MSE) decays as M^{-4} for an M-element frame, which is consistent with previous results on optimal decay of MSE. Reconstruction using the canonical dual frame is also studied, and several results relate properties of the analysis frame to whether linear reconstruction techniques provide consistent reconstructions.

© 2010 Elsevier Inc. All rights reserved.

1. Introduction

Redundant representations obtained with frames are playing an ever-expanding role in signal processing due to design flexibility and other desirable properties [1,2]. One such favorable property is robustness to additive noise [3]. This robustness, carried over to quantization noise (without regard to whether it is random or signal-independent), explains the success of both ordinary oversampled analog-to-digital conversion (ADC) and Σ–Δ ADC with the canonical linear reconstruction. But the combination of frame expansions with scalar quantization is considerably more interesting and intricate because boundedness of quantization noise can be exploited in reconstruction [4–12] and frames and quantizers can be designed jointly to obtain favorable performance [13].

This paper introduces a new use of finite frames in vector quantization: *frame permutation quantization* (FPQ). In FPQ, permutation source coding (PSC) [14,15] is applied to a frame expansion of a vector. This means that the vector is represented by a partial ordering of the frame coefficients (Variant I) or by signs of the frame coefficients that are larger than some threshold along with a partial ordering of the absolute values of the significant coefficients (Variant II). FPQ provides

a space partitioning that can be combined with additional signal constraints or prior knowledge to generate a variety of vector quantizers.

Beyond the explication of the basic ideas in FPQ, the focus of this paper is on how—in analogy to works cited above—there are several decoding procedures that can sensibly be used with the encoding of FPQ. First, we consider using the ordinary PSC decoding for the frame coefficients followed by linear synthesis with the canonical dual; from the perspective of frame theory, this is the natural way to reconstruct. For this, we find conditions on the frame used in FPQ that relate to whether the canonical reconstruction is consistent. Second, taking a geometric approach based on imposing consistency yields instead optimization-based algorithms. Third, algorithms with lower complexity can have similar performance by recursively imposing consistency only locally [8,12].

There are two distinct ways to measure the performance of FPQ, and these correspond to different potential uses of FPQ: data compression and data acquisition. The accuracy of signal representation—here measured by mean-squared error (MSE)—is important in either case. For data compression, accuracy is traded off against a coding rate (bits per sample). The standard alternative is scalar quantization, and for low delay and complexity, one considers moderate signal dimensions. It is remarkable that introducing redundancy through a frame expansion can improve compression, and we find that it does so only when the redundancy is low. For data acquisition, accuracy is traded off against the number of samples collected (number of frame elements). Sensors that operate at low power and high speed by outputting orderings of signal levels rather than absolute levels have been demonstrated and are a subject of renewed interest [16,17]. By showing that the MSE can decay quickly as a function of the number of samples collected, we may encourage the further development of such sensors. Here computational complexity of reconstruction is more important because the data are recoded prior to storage or transmission. This is in close analogy to oversampling in analog-to-digital conversion, which is ubiquitous even though it is not advantageous in terms of accuracy as a function of bit rate unless there is recoding at or near Nyquist rate [18,19]. Note also that for both historical and practical reasons, data compression is typically studied for random vectors while data acquisition is studied for nonrandom vectors within some bounded set [20]. This paper mixes Bayesian and non-Bayesian formulations accordingly.

The paper is organized as follows: Before formal introduction to frame expansions, permutation source codes, or their combination, Section 2 provides a preview of the geometry of FPQ. This serves both to contrast with ordinary scalar-quantized frame expansions and to see the effect of frame redundancy. Section 3 provides the requisite background by reviewing PSCs, frames, and scalar-quantized frame expansions. Section 4 formally defines FPQ, emphasizing constraints that are implied by the representation and hence must be satisfied for consistent reconstruction. Section 4 also provides reconstruction algorithms based on applying the constraints for consistent reconstruction globally or locally. The results on choices of frames in FPQ appear in Section 5. These are necessary and sufficient conditions on frames for linear reconstructions to be consistent. Section 6 provides numerical results that demonstrate improvement in operational distortion–rate compared to ordinary PSC and optimal decay of distortion as a function of the number of samples. Proofs of the main results are given in Section 7. Preliminary results on FPQ were mentioned briefly in [21].

2. Preview through $\mathbb{R}^2$ geometry

Consider the quantization of $x \in \mathbb{R}^N$, where we restrict attention to $N = 2$ in this section but later allow any finite N. The uniform scalar quantization of x partitions $\mathbb{R}^N$ in a trivial way, as shown in Fig. 1(a). (An arbitrary segment of the plane is shown.) If over a domain of interest each component is divided into K intervals, a partition with K^N cells is obtained.

A way to increase the number of partition cells without increasing the scalar quantization resolution is to use a frame expansion. A conventional quantized frame expansion is obtained by scalar quantization of $y = Fx$, where $F \in \mathbb{R}^{M \times N}$ with $M \geqslant N$. Keeping the resolution K fixed, the partition now has K^M cells. An example with $M = 6$ is shown in Fig. 1(d). Each frame element ϕ_k (transpose of row of F) induces a *hyperplane wave partition* [22]: a partition formed by equally-spaced $(N-1)$-dimensional hyperplanes normal to ϕ_k. The overall partition has M hyperplane waves and is spatially uniform. A spatial shift invariance can be ensured formally by the use of subtractively dithered quantizers [23].

A Variant I PSC represents x just by which permutation of the components of x puts the components in descending order. In other words, only whether $x_1 > x_2$ or $x_2 > x_1$ is specified.[1] The resulting partition is shown in Fig. 1(b). A Variant II PSC specifies (at most) the signs of the components of x_1 and x_2 and whether $|x_1| > |x_2|$ or $|x_2| > |x_1|$. The corresponding partitioning of the plane is shown in Fig. 1(c), with the vertical line coming from the sign of x_1, the horizontal line coming from the sign of x_2, and the diagonal lines from $|x_1| \gtrless |x_2|$.

While low-dimensional diagrams are often inadequate in explaining PSC, several key properties are illustrated. The partition cells are (unbounded) convex cones, giving special significance to the origin and a lack of spatial shift invariance. The unboundedness of cells implies that some additional knowledge, such as a bound on $\|x\|$ or a probabilistic distribution on x, is needed to compute good estimates. At first this may seem extremely different from ordinary scalar quantization or scalar-quantized frame expansions, but those techniques also require some prior knowledge to allow the quantizer outputs

---

[1] The boundary case of $x_1 = x_2$ can be handled arbitrarily in practice and safely ignored in the analysis. When the source vector has an absolutely continuous distribution, the boundary affects neither the rate nor the distortion. For an optimal quantizer, the boundaries will have zero probability even if the source has a discrete component [24, p. 355].
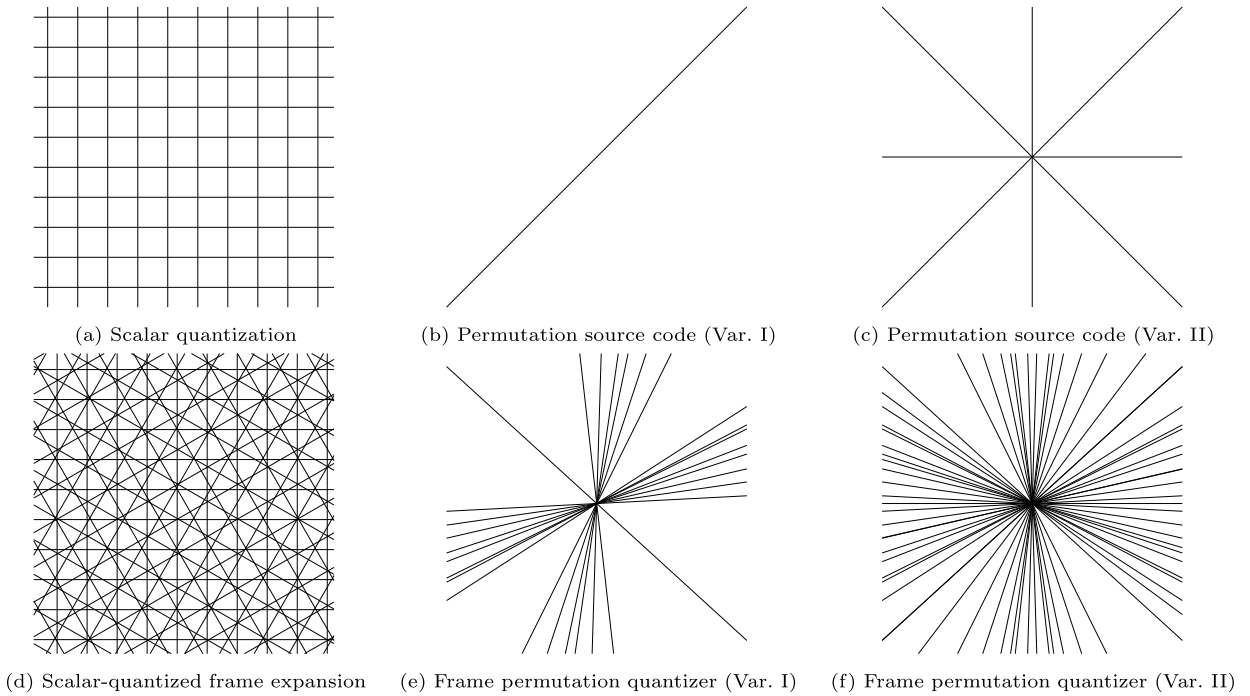
(a) Scalar quantization   (b) Permutation source code (Var. I)   (c) Permutation source code (Var. II)

(d) Scalar-quantized frame expansion   (e) Frame permutation quantizer (Var. I)   (f) Frame permutation quantizer (Var. II)

Fig. 1. Partition diagrams for $x \in \mathbb{R}^2$. (a) Scalar quantization. (b) Permutation source code, Variant I. (c) Permutation source code, Variant II. (Both permutation source codes have $n_1 = n_2 = 1$.) (d) Scalar-quantized frame expansion with $M = 6$ coefficients (real harmonic tight frame). (e) Frame permutation quantizer, Variant I. (f) Frame permutation quantizer, Variant II. (Both frame permutation quantizers have $M = 6$, $m_1 = m_2 = \cdots = m_6 = 1$, and the same random frame.)

to be represented with finite numbers of bits. We also see that the dimension N determines the maximum number of cells ($N!$ for Variant I and $2^N N!$ for Variant II); there is no parameter analogous to scalar quantization step size that allows arbitrary control of the resolution.

To get a finer partition without changing the dimension N, we can again employ a frame expansion. With $y = Fx$ as before, PSC of y gives more relative orderings with which to represent x. If ϕ_j and ϕ_k are frame elements (transposes of rows of F) then $\langle x, \phi_j \rangle \gtrless \langle x, \phi_k \rangle$ is $\langle x, \phi_j - \phi_k \rangle \gtrless 0$ by linearity of the inner product, so every pair of frame elements can give a condition on x. An example of a partition obtained with Variant I and $M = 6$ is shown in Fig. 1(e). There are many more cells than in Fig. 1(b). Similarly, Fig. 1(f) shows a Variant II example. The cells are still (unbounded) convex cones. If additional information such as $\|x\|$ or an affine subspace constraint (not passing through the origin) is known, x can be specified arbitrarily closely by increasing M.

3. Background

Having illustrated the basic idea of PSC and our generalization using frames to provide resolution control, we now formalize the background material. We assume throughout fixed-rate coding and the conventional squared-error fidelity criterion $\|x - \hat{x}\|^2$ between source x and reproduction $\hat{x}$. Some statements—especially those pertaining to data compression—assume a known source distribution over which performance is measured in expectation. Most statements for data acquisition with $M \to \infty$ apply pointwise over x.

3.1. Vector quantization

A vector quantizer is a mapping from an input $x \in \mathbb{R}^N$ to a *codeword* $\hat{x}$ from a finite *codebook* $\mathcal{C}$. Without loss of generality, a vector quantizer can be seen as the combination of an *encoder*

$$\alpha : \mathbb{R}^N \to \mathcal{I}$$

and a *decoder*

$$\beta : \mathcal{I} \to \mathbb{R}^N,$$

where $\mathcal{I}$ is a finite index set. The encoder partitions $\mathbb{R}^N$ into $|\mathcal{I}|$ regions or *cells* $\{\alpha^{-1}(i)\}_{i \in \mathcal{I}}$, and the decoder assigns a *reproduction value* to each cell. Examples of partitions are given in Fig. 1. For the quantizer to output R bits per component, we have $|\mathcal{I}| = 2^{NR}$.

For any codebook (i.e., any β), the encoder α that minimizes $\|x - \hat{x}\|^2$ maps x to the nearest element of the codebook. The partition is thus composed of convex cells. Since the cells are convex, reproduction values are optimally within the corresponding cells–whether to minimize mean-squared error distortion, maximum squared error, or some other reasonable function of squared error. To minimize maximum squared error, reproduction values should be at centers of cells; to minimize expected distortion, they should be at centroids of cells. Reproduction values being within corresponding cells is formalized as *consistency*:

Definition 3.1. The reconstruction $\hat{x} = \beta(\alpha(x))$ is called a *consistent reconstruction of x* when $\alpha(x) = \alpha(\hat{x})$ (or equivalently $\beta(\alpha(\hat{x})) = \hat{x}$). The decoder β is called *consistent* when $\beta(\alpha(x))$ is a consistent reconstruction of x for all x.

In practice, the pair (α, β) usually does not minimize any desired distortion criterion for a given codebook size because the optimal mappings are hard to design and hard to implement [24]. The mappings are commonly designed subject to certain structural constraints, and β may not even be consistent for α [4,6].

3.2. Permutation source codes

A permutation source code is a vector quantizer with the defining characteristic that codewords are related through permutations and, possibly, sign changes. Permutation codes were originally introduced as channel codes by Slepian [25]. They were then applied to a specific source coding problem, through the duality between source encoding and channel decoding, by Dunn [14] and developed in greater generality by Berger et al. [15,26,27]. Permutation codes are generated by the group action of a permutation group and are thus examples of group codes [28].

3.2.1. Definitions

There are two variants of permutation codes:

Variant I: Here codewords are related through permutations, without sign changes. Let $\mu_1 > \mu_2 > \cdots > \mu_K$ be real numbers, and let $n_1, n_2, \ldots, n_K$ be positive integers that sum to N (an *(ordered) composition* of N). The *initial codeword* of the codebook $\mathcal{C}$ has the form

$$\hat{x}_{\mathrm{init}} = (\underbrace{\mu_1, \ldots, \mu_1}_{n_1}, \underbrace{\mu_2, \ldots, \mu_2}_{n_2}, \ldots, \underbrace{\mu_K, \ldots, \mu_K}_{n_K}), \tag{1}$$

where each μ_i appears n_i times. When $\hat{x}_{\mathrm{init}}$ has this form, we call it *compatible with* $(n_1, n_2, \ldots, n_K)$. The codebook is the set of all distinct permutations of $\hat{x}_{\mathrm{init}}$. The number of codewords in $\mathcal{C}$ is thus given by the multinomial coefficient

$$L_{\mathrm{I}} = \frac{N!}{n_1! n_2! \cdots n_K!}. \tag{2a}$$

The permutation structure of the codebook enables low-complexity nearest-neighbor encoding [15]: map x to the codeword $\hat{x}$ whose components have the same order as x; in other words, replace the n_1 largest components of x with μ_1, the n_2 next-largest components of x with μ_2, and so on.

Variant II: Here codewords are related through permutations and sign changes. Let $\mu_1 > \mu_2 > \cdots > \mu_K \geqslant 0$ be nonnegative real numbers, and let $(n_1, n_2, \ldots, n_K)$ be a composition of N. The initial codeword has the same form as in (1), and the codebook now consists of all distinct permutations of $\hat{x}_{\mathrm{init}}$ with each possible sign for each nonzero component. The number of codewords in $\mathcal{C}$ is thus given by

$$L_{\mathrm{II}} = 2^h \frac{N!}{n_1! n_2! \cdots n_K!}, \tag{2b}$$

where $h = N$ if $\mu_K > 0$ and $h = N - n_K$ if $\mu_K = 0$.

Nearest-neighbor encoding for Variant II PSCs can be implemented as follows [15]: map x to the codeword $\hat{x}$ whose components have the same order in absolute value and match the signs of corresponding components of x. Since the complexity of sorting a vector of length N is $O(N \log N)$ operations, the encoding complexity for either PSC variant is much lower than with an unstructured source code and only $O(\log N)$ times higher than scalar quantization.

With the codebook sizes given in (2), the per-component rate is defined as

$$R = N^{-1} \log_2 L. \tag{3}$$

Under certain symmetry conditions on the source distribution, all codewords are equally likely so the rate cannot be reduced by entropy coding. This generation of fixed-rate output–avoiding the possibility of buffer overflow associated with entropy coding of the highly nonequiprobable outputs of a quantizer [29]–is a known advantage of PSCs [15]. An efficient enumeration of permutations, to generate a binary representation, is described in [30].

3.2.2. Partition properties

For both historical reasons and to match the conventional approach to vector quantization, PSCs were defined above in terms of a codebook structure, and the codebook structure led to an encoding procedure. Note that we may now examine the partitions induced by PSCs separately from the particular codebooks for which they are nearest-neighbor partitions.

The partition induced by a Variant I PSC is completely determined by the composition $(n_1, n_2, \ldots, n_K)$. Specifically, the encoding mapping can index the permutation P that places the n_1 largest components of x in the first n_1 positions (without changing the order within those n_1 components), the n_2 next-largest components of x in the next n_2 positions, and so on; the μ_is are actually immaterial. This encoding is placing all source vectors x such that Px is n-descending in the same partition cell, defined as follows.

Definition 3.2. Given a composition $n = (n_1, n_2, \ldots, n_K)$ of N, a vector in $\mathbb{R}^N$ is called *n-descending* if its n_1 largest entries are in the first n_1 positions, its n_2 next-largest components are in the next n_2 positions, etc.

The property of being n-descending is to be descending up to the arbitrariness specified by the composition n.

Because this is nearest-neighbor encoding for *some* codebook, the partition cells must be convex. Furthermore, multiplying x by any nonnegative scalar does not affect the encoding, so the cells are convex cones. (This was discussed and illustrated in Section 2.) We develop a convenient representation for the partition in Section 4.

The situation is only slightly more complicated for Variant II PSCs. The partition is determined by the composition $(n_1, n_2, \ldots, n_K)$ and whether or not the signs of the smallest-magnitude components should be encoded (whether $\mu_K = 0$, in the codebook-centric view).

The PSC literature has mostly emphasized the design of PSCs for sources with i.i.d. components. But as developed in Section 4, the simple structured encoding of PSCs could be combined with unconventional decoding techniques for other sources. The possible suitability of PSCs for sources with unknown or time-varying statistics has been previously observed [15].

3.2.3. Codebook optimization

With the encoding procedure now fixed, let us turn to the decoder (or codebook) design. For this we assume that x is random and that the components of x are i.i.d.

Let $\xi_1 \geqslant \xi_2 \geqslant \cdots \geqslant \xi_N$ denote the order statistics of random vector $x = (x_1, \ldots, x_N)$ and $\eta_1 \geqslant \eta_2 \geqslant \cdots \geqslant \eta_N$ denote the order statistics of random vector $|x| \triangleq (|x_1|, \ldots, |x_N|)$.[2] For a given initial codeword $\hat{x}_{\text{init}}$, the per-letter distortion of optimally-encoded Variant I and Variant II PSCs are given by

$$D_{\text{I}} = N^{-1} E\left[\sum_{i=1}^{K} \sum_{\ell \in \mathcal{I}_i} (\xi_\ell - \mu_i)^2 \right] \tag{4a}$$

and

$$D_{\text{II}} = N^{-1} E\left[\sum_{i=1}^{K} \sum_{\ell \in \mathcal{I}_i} (\eta_\ell - \mu_i)^2 \right], \tag{4b}$$

where $\mathcal{I}_i$s are the sets of indexes generated by the composition:

$$\mathcal{I}_1 = \{1, 2, \ldots, n_1\}, \tag{5a}$$

$$\mathcal{I}_i = \left\{ \left(\sum_{k=1}^{i-1} n_k \right) + 1, \left(\sum_{k=1}^{i-1} n_k \right) + 2, \ldots, \left(\sum_{k=1}^{i} n_k \right) \right\}, \quad i \geqslant 2. \tag{5b}$$

These distortions can be deduced simply by examining which components of x are mapped to which elements of $\hat{x}_{\text{init}}$.

Optimization of (4a) and (4b) over both $\{n_i\}_{i=1}^{K}$ and $\{\mu_i\}_{i=1}^{K}$ subject to (3) is difficult, partly due to the integer constraint of the composition. However, given a composition $(n_1, n_2, \ldots, n_K)$, the optimal initial codeword can be determined easily from the means of the order statistics. In particular, the optimal $\{\mu_i\}_{i=1}^{K}$ of Variant I and Variant II PSCs are given by

$$\mu_i = n_i^{-1} \sum_{\ell \in \mathcal{I}_i} E[\xi_\ell], \quad \text{for Variant I,} \tag{6a}$$

and

$$\mu_i = n_i^{-1} \sum_{\ell \in \mathcal{I}_i} E[\eta_\ell], \quad \text{for Variant II.} \tag{6b}$$

---

[2] For consistency with earlier literature on PSCs, we are reversing the usual sorting of order statistics [31].

The analysis of [26] shows that when N is large, the optimal composition gives performance equal to optimal entropy-constrained scalar quantization (ECSQ) of x. Performance does not strictly improve with increasing N; permutation codes outperform ECSQ for certain combinations of block size and rate [32].

3.3. Frame definitions and classifications

The theory of finite-dimensional frames is often developed for a Hilbert space $\mathbb{C}^N$ of complex vectors. In this paper, we use frame expansions only for quantization using PSCs, which rely on order relations of real numbers. Therefore we limit ourselves to real finite frames. We maintain the Hermitian transpose notation $*$ where a transpose would suffice because this makes several expressions have familiar appearances.

The Hilbert space of interest is $\mathbb{R}^N$ equipped with the standard inner product (dot product),

$$\langle x, y \rangle = x^T y = \sum_{k=1}^{N} x_k y_k,$$

for $x = [x_1, x_2, \ldots, x_N]^T \in \mathbb{R}^N$ and $y = [y_1, y_2, \ldots, y_N]^T \in \mathbb{R}^N$. The norm of a vector x is naturally induced from the inner product,

$$\|x\| = \sqrt{\langle x, x \rangle}.$$

Definition 3.3. (See [3].) A set of N-dimensional vectors, $\Phi = \{\phi_k\}_{k=1}^{M} \subset \mathbb{R}^N$, is called a *frame* if there exist a lower frame bound, $A > 0$, and an upper frame bound, $B < \infty$, such that

$$A\|x\|^2 \leqslant \sum_{k=1}^{M} \left| \langle x, \phi_k \rangle \right|^2 \leqslant B\|x\|^2, \quad \text{for all } x \in \mathbb{R}^N. \tag{7a}$$

The matrix $F \in \mathbb{R}^{M \times N}$ with kth row equal to ϕ_k^* is called the *analysis frame operator*. F and Φ will be used interchangeably to refer to a frame. Equivalent to (7a) in matrix form is

$$A I_N \leqslant F^* F \leqslant B I_N, \tag{7b}$$

where I_N is the $N \times N$ identity matrix.

The lower bound in (7) implies that Φ spans $\mathbb{R}^N$; thus a frame must have $M \geqslant N$. It is therefore reasonable to call the ratio $r = M/N$ the *redundancy* of the frame. A frame is called a *tight frame* if the frame bounds can be chosen to be equal. A frame is an *equal-norm frame* if all of its vectors have the same norm. If an equal-norm frame is normalized to have all vectors of unit norm, we call it a *unit-norm frame* (or sometimes *normalized frame* or *uniform frame*). For a unit-norm frame, it is easy to verify that $A \leqslant r \leqslant B$. Thus, a unit-norm tight frame (UNTF) must satisfy $A = r = B$ and

$$F^* F = r I_N. \tag{8}$$

Naimark's theorem [33] provides an efficient way to characterize the class of equal-norm tight frames: a set of vectors is an equal-norm tight frame if and only if it is the orthogonal projection (up to a scale factor) of an orthonormal basis of an ambient Hilbert space on to some subspace.[3] As a consequence, deleting the last $(M - N)$ columns of the (normalized) discrete Fourier transform (DFT) matrix in $\mathbb{C}^{M \times M}$ yields a particular subclass of UNTFs called *(complex) harmonic tight frames* (HTFs). One can adapt this derivation to construct *real* HTFs [34], which are always UNTFs, as follows.

Definition 3.4. The real harmonic tight frame of M vectors in $\mathbb{R}^N$ is defined for even N by

$$\phi_{k+1}^* = \sqrt{\frac{2}{N}} \left[\cos \frac{k\pi}{M}, \cos \frac{3k\pi}{M}, \ldots, \cos \frac{(N-1)k\pi}{M}, \sin \frac{k\pi}{M}, \sin \frac{3k\pi}{M}, \ldots, \sin \frac{(N-1)k\pi}{M} \right] \tag{9a}$$

and for odd N by

$$\phi_{k+1}^* = \sqrt{\frac{2}{N}} \left[\frac{1}{\sqrt{2}}, \cos \frac{2k\pi}{M}, \cos \frac{4k\pi}{M}, \ldots, \cos \frac{(N-1)k\pi}{M}, \sin \frac{2k\pi}{M}, \sin \frac{4k\pi}{M}, \ldots, \sin \frac{(N-1)k\pi}{M} \right], \tag{9b}$$

where $k = 0, 1, \ldots, M - 1$. The *modulated harmonic tight frames* are defined by

$$\psi_k = \gamma(-1)^k \phi_k, \quad \text{for } k = 1, 2, \ldots, M, \tag{10}$$

where $\gamma = 1$ or $\gamma = -1$ (fixed for all k).

---

[3] The theorem holds for a general separable Hilbert space of possibly infinite dimension.

HTFs can be viewed as the result of a group of orthogonal operators acting on one generating vector [2]. This property has been generalized in [35,36] under the name *geometrically-uniform frames* (GUFs). Note that a GUF is a special case of a group code as developed by Slepian [25,28]. An interesting connection between PSCs and GUFs is that under certain conditions, a PSC codebook is a GUF with generating vector $\hat{x}_{init}$ and the generating group action provided by all permutation matrices [37].

Classification of frames is often up to some unitary equivalence [34]. Adopting the terminology of Holmes and Paulsen [38], we say two frames in $\mathbb{R}^N$, $\Phi = \{\phi_k\}_{k=1}^M$ and $\Psi = \{\psi_k\}_{k=1}^M$, are

(i) Type I equivalent if there is an orthogonal matrix U such that $\psi_k = U\phi_k$ for all k;

(ii) Type II equivalent if there is a permutation $\sigma(\cdot)$ on $\{1, 2, \ldots, M\}$ such that $\psi_k = \phi_{\sigma(k)}$ for all k; and

(iii) Type III equivalent if there is a *sign function* in k, $\delta(k) = \pm 1$ such that $\psi_k = \delta(k)\phi_k$ for all k.

It will be evident that FPQ performance is always invariant to Type II equivalence; invariant to Type I equivalence when the source distribution is rotationally invariant; and invariant to Type III equivalence under Variant II but not, in general, under Variant I.

It is important to note that for $M = N + 1$ there is exactly one equivalence class of UNTFs [34, Thm. 2.6]. Since HTFs are always UNTFs, the following property follows directly from [34, Thm. 2.6].

Proposition 3.5. *Assume that $M = N + 1$, and $\Phi = \{\phi_k\}_{k=1}^M \subset \mathbb{R}^N$ is the real HTF. Then every UNTF $\Psi = \{\psi_k\}_{k=1}^M$ can be written as*

$$\psi_k = \delta(k)U\phi_{\sigma(k)}, \quad for\ k = 1, 2, \ldots, M, \tag{11}$$

where $\delta(k) = \pm 1$ is some sign function in k, U is some orthogonal matrix, and $\sigma(\cdot)$ is some permutation on the index set $\{1, 2, \ldots, M\}$.

Another important subclass of UNTFs is defined as follows:

Definition 3.6. (See [39,40].) A UNTF $\Phi = \{\phi_k\}_{k=1}^M \subset \mathbb{R}^N$ is called an *equiangular tight frame* (ETF) if there exists a constant a such that $|\langle \phi_\ell, \phi_k \rangle| = a$ for all $1 \leqslant \ell < k \leqslant M$.

ETFs are sometimes called *optimal Grassmannian frames* or *2-uniform frames*. They prove to have rich application in communications, coding theory, and sparse approximation [38,39,41]. For a general pair (M, N), the existence and constructions of such frames is not fully understood. Partial answers can be found in [40,42,43].

In our analysis of FPQ, we will find that *restricted* ETFs—where the absolute value constraint can be removed from Definition 3.6—play a special role. In matrix view, a restricted ETF satisfies $F^*F = rI_N$ and $FF^* = (1 - a)I_M + aJ_M$, where J_M is the all-1s matrix of size $M \times M$. The following proposition specifies the restricted ETFs for the codimension-1 case.

Proposition 3.7. *For $M = N + 1$, the family of all restricted ETFs is constituted by the Type I and Type II equivalents of modulated HTFs.*

Proof. See Section 7.1. □

The following property of modulated HTFs in the $M = N + 1$ case will be very useful.

Proposition 3.8. *If $M = N + 1$ then a modulated harmonic tight frame is a zero-sum frame, i.e., each column of the analysis frame operator F sums to zero.*

Proof. We only consider the case when N is even; the N odd case is similar. For each $\ell \in \{1, 2, \ldots, N\}$, let ϕ_k^ℓ denote the ℓth component of vector ϕ_k and let $S_\ell = \sum_{k=1}^M \phi_k^\ell$ denote the sum of the entries in column ℓ of matrix F.

For $1 \leqslant \ell \leqslant N/2$, using Euler's formula, we have

$$S_\ell = \pm\sqrt{\frac{2}{N}} \sum_{k=0}^{M-1} (-1)^k \cos\frac{(2\ell - 1)k\pi}{M} \propto \sum_{k=0}^{M-1} e^{jk\pi}\left[e^{j(2\ell-1)k\pi/M} + e^{-j(2\ell-1)k\pi/M}\right]$$

$$= \sum_{k=0}^{M-1} e^{j\pi((2\ell-1)/M+1)k} + \sum_{k=0}^{M-1} e^{-j\pi((2\ell-1)/M+1)k} = \frac{1 - e^{j\pi(2\ell+M-1)}}{1 - e^{j\pi((2\ell-1)/M+1)}} + \frac{1 - e^{-j\pi(2\ell+M-1)}}{1 - e^{-j\pi((2\ell-1)/M+1)}} = 0, \tag{12}$$

where (12) follows from the fact that $2\ell + M - 1 = 2\ell + N$ is an even integer.

For $N/2 < \ell \leqslant N$, we can show that $S_\ell = 0$ similarly, and so the proposition is proved. □

Algorithm 1: Consistent reconstruction from scalar-quantized frame expansion.

Inputs:	Analysis frame operator F, quantization step size Δ, and quantized frame expansion $\hat{y}$
Output:	Estimate $\hat{x}$ consistent with $\hat{y}$ and as far from the partition boundaries as possible

1. Let $A = \begin{bmatrix} F & 1_{M\times 1} \\ -F & 1_{M\times 1} \end{bmatrix}$ and $b = \begin{bmatrix} \hat{y} \\ -\hat{y} \end{bmatrix}$.
   (Consistency as in (14) is expressed as $A\begin{bmatrix} x \\ 0 \end{bmatrix} \leqslant b$.)
2. Let $c = \begin{bmatrix} 0_{N\times 1} \\ -1 \end{bmatrix}$.
3. Use a linear programming method to minimize $c^T \begin{bmatrix} x \\ \delta \end{bmatrix}$ subject to $A\begin{bmatrix} x \\ \delta \end{bmatrix} \leqslant b$.
   Return the first N components of the minimizer as $\hat{x}$.

3.4. Reconstruction from frame expansions

A central use of frames is to formalize the reconstruction of $x \in \mathbb{R}^N$ from the frame expansion $y_k = \langle x, \phi_k \rangle$, $k = 1, 2, \ldots, M$, or estimation of x from degraded versions of the frame expansion. Using the analysis frame operator we have $y = Fx$, and (7) implies the existence of at least one linear *synthesis operator* G such that $GF = I_N$. A frame with analysis frame operator G^* is then said to be *dual* to Φ.

The frame condition (7) also implies that F^*F is invertible, so the Moore–Penrose inverse (pseudo-inverse) of the frame operator

$$F^\dagger = \left(F^*F\right)^{-1} F^*$$

exists and is a valid synthesis operator. Using the pseudo-inverse for reconstruction has several important properties including an optimality for mean-squared error (MSE) under assumptions of uncorrelated zero-mean additive noise and linear synthesis [3, Sect. 3.2]. This follows from the fact that $FF^\dagger$ is an orthogonal projection from $\mathbb{R}^M$ onto the subspace $F(\mathbb{R}^N)$, the range of F. Because of this special role, reconstruction using $F^\dagger$ is called *canonical reconstruction* and the corresponding frame is called the *canonical dual*. In this paper, we use the term *linear reconstruction* for reconstruction using an arbitrary linear operator.

When y is quantized to $\hat{y}$, it is possible for the quantization noise $\hat{y} - y$ to have mean zero and uncorrelated components; this occurs with subtractive dithered quantization [23] or under certain asymptotics [44]. In this case, the optimality of canonical reconstruction holds. However, it should be noted that even with these restrictions, canonical reconstruction is optimal *only amongst linear reconstructions.*

When nonlinear construction is allowed, quantization noise may behave fundamentally differently than other additive noise. The key is that a quantized value gives hard constraints that can be exploited in reconstruction. For example, suppose that $\hat{y}$ is obtained from y by rounding each element to the nearest multiple of a quantization step size Δ. Then knowledge of $\hat{y}_k$ is equivalent to knowing

$$y_k \in \left[\hat{y}_k - \frac{1}{2}\Delta, \hat{y}_k + \frac{1}{2}\Delta\right]. \tag{13}$$

Geometrically, $\langle x, \phi_k \rangle = \hat{y}_k - \frac{1}{2}\Delta$ and $\langle x, \phi_k \rangle = \hat{y}_k + \frac{1}{2}\Delta$ are hyperplanes perpendicular to ϕ_k, and (13) expresses that x must lie between these hyperplanes; this may be visualized as one pair of parallel lines in Fig. 1(d). Using the upper and lower bounds on all M components of y, the constraints on x imposed by $\hat{y}$ are readily expressed as (see [6])

$$\begin{bmatrix} F \\ -F \end{bmatrix} x \leqslant \begin{bmatrix} \frac{1}{2}\Delta + \hat{y} \\ \frac{1}{2}\Delta - \hat{y} \end{bmatrix}, \tag{14}$$

where the inequalities are elementwise. For example, all $2M$ constraints specify a single cell in Fig. 1(d). This formulation inspires Algorithm 1, which is a modification of [6, Table I] using the principle of maximizing slackness of inequalities that was also implemented in [8]. Section 4.3 presents analogues to (14) and Algorithm 1 for FPQ.

The cost of Algorithm 1 may be prohibitive if M is large. In particular, Algorithm 1 uses a linear program with $N + 1$ variables and $2M$ constraints, and solving this has cost that is superlinear in M. One way to reduce the cost to linear in M is to use each of M quantized coefficients only once and in a computation with constant cost. Algorithm 2 uses each constraint (13) once, recursively, in isolation of all other constraints. It uses y_k by orthogonally projecting a running estimate $\hat{x}_{k-1}$ to the set consistent with (13). Remarkably, even though the final estimate is generally not consistent with all M constraints of the form (13), the optimal $\Theta(M^{-2})$ decay of $\|x - \hat{x}\|^2$ as a function of the number of coefficients M can be attained under appropriate technical conditions [8,12].

4. Frame permutation quantization

With background material on permutation source codes and finite frames in place, we are now prepared to formally introduce frame permutation quantization. FPQ is simply PSC applied to a frame expansion.

Algorithm 2: Recursive estimation from scalar-quantized frame coefficient sequence.

Inputs:	Analysis frame sequence $\{\phi_j\}_{j=1}^M$, quantization step size Δ, and quantized coefficients $\{\hat{y}_j\}_{j=1}^M$
Output:	Estimate $\hat{x}$

1. Let $\hat{x}_0 = 0_{N \times 1}$ and let $k = 1$.
2. If $\langle \hat{x}_{k-1}, \phi_k \rangle < \hat{y}_k - \frac{1}{2}\Delta$, let $\hat{x}_k = \hat{x}_{k-1} + (\hat{y}_k - \frac{1}{2}\Delta - \langle \hat{x}_{k-1}, \phi_k \rangle)\phi_k / \|\phi_k\|^2$;
   else if $\langle \hat{x}_{k-1}, \phi_k \rangle > \hat{y}_k + \frac{1}{2}\Delta$, let $\hat{x}_k = \hat{x}_{k-1} + (\hat{y}_k + \frac{1}{2}\Delta - \langle \hat{x}_{k-1}, \phi_k \rangle)\phi_k / \|\phi_k\|^2$;
   else let $\hat{x}_k = \hat{x}_{k-1}$.
3. If $k = M$, return $\hat{x}_k$; else increment k and go to step 2.

4.1. Encoder definition and canonical decoding

Definition 4.1. A *frame permutation quantizer* with analysis frame $F \in \mathbb{R}^{M \times N}$, composition $m = (m_1, m_2, \ldots, m_K)$, and initial codeword $\hat{y}_{\text{init}}$ compatible with m encodes $x \in \mathbb{R}^N$ by applying a permutation source code with composition m and initial codeword $\hat{y}_{\text{init}}$ to Fx. The *canonical decoding* gives $\hat{x} = F^\dagger \hat{y}$, where $\hat{y}$ is the PSC reconstruction of y.

The two variants of PSCs yield two variants of FPQ. We sometimes use the triple $(F, m, \hat{y}_{\text{init}})$ along with a specification of Variant I or Variant II to refer to such an FPQ.

For Variant I, the result of the encoding can be expressed as a permutation P from the permutation matrices of size M. The permutation is such that PFx is m-descending. For uniqueness in the representation P chosen from the set of permutation matrices, we can specify that the first m_1 components of Py are kept in the same order as they appeared in y, the next m_2 components of Py are kept in the same order as they appeared in y, etc. Then P is in a subset $\mathcal{G}(m)$ of the $M \times M$ permutation matrices and

$$\big|\mathcal{G}(m)\big| = \frac{M!}{m_1! m_2! \cdots m_K!}. \tag{15}$$

Notice that, analogous to the discussion in Section 3.2.2, the encoding uses the composition m but not the initial codeword $\hat{y}_{\text{init}}$. The PSC reconstruction of y is $P^{-1}\hat{y}_{\text{init}}$, so the canonical decoding gives $F^\dagger P^{-1}\hat{y}_{\text{init}}$.

For Variant II, we will sidestep the differences between the $\mu_K = 0$ and $\mu_K \neq 0$ cases in Section 3.2 by specifying that the signs of the m_K smallest-magnitude components of Fx are not encoded and $m_K = 0$ is allowed. Now the result of encoding can be expressed similarly as $P \in \mathcal{G}(m)$ along with a diagonal matrix V with ± 1 on its diagonal. These matrices are selected such that the elementwise absolute values of $VPFx$ are m-descending and also the first $M - m_K$ entries of $VPFx$ are positive. The last m_K diagonal entries of V do not affect the encoding and are set to $+1$. Thus V is in a subset $\mathcal{Q}(m)$ of the $M \times M$ sign-changing matrices and

$$\big|\mathcal{Q}(m)\big| = 2^{M-m_K}. \tag{16}$$

The PSC reconstruction of y is $P^{-1}V^{-1}\hat{y}_{\text{init}}$, so the canonical decoding gives $F^\dagger P^{-1}V^{-1}\hat{y}_{\text{init}}$.

The sizes of the sets $\mathcal{G}(m)$ and $\mathcal{G}(m) \times \mathcal{Q}(m)$ are analogous to the codebook sizes in (2), and the per-component rates of FPQ are thus defined as

$$R_{\text{I}} = N^{-1} \log_2 \frac{M!}{m_1! m_2! \cdots m_K!}, \quad \text{for Variant I,} \tag{17a}$$

and

$$R_{\text{II}} = N^{-1}\left(M - m_K + \log_2 \frac{M!}{m_1! m_2! \cdots m_K!} \right), \quad \text{for Variant II.} \tag{17b}$$

4.2. Expressing consistency constraints

Suppose FPQ encoding of $x \in \mathbb{R}^N$ with frame $F \in \mathbb{R}^{M \times N}$, composition $m = (m_1, m_2, \ldots, m_K)$, and initial codeword $\hat{y}_{\text{init}}$ compatible with m results in permutation $P \in \mathcal{G}(m)$ (and, in the case of Variant II, $V \in \mathcal{Q}(m)$) as described in Section 4.1. We would like to express constraints on x that are specified by $(F, m, \hat{y}_{\text{init}}, P)$ (or $(F, m, \hat{y}_{\text{init}}, P, V)$). This will provide an explanation of the partitions induced by FPQ and lead to reconstruction algorithms in Section 4.3.

Knowing that a vector is m-descending is a specification of many inequalities. Recall the definitions of the index sets generated by a composition given in (5), and use the same notation with n_ks replaced by m_ks. Then z being m-descending implies that for any $i < j$,

$$z_k \geqslant z_\ell, \quad \text{for every } k \in \mathcal{I}_i \text{ and } \ell \in \mathcal{I}_j.$$

By transitivity, considering every $i < j$ gives redundant inequalities. Taking only $j = i + 1$, we obtain a full description

$$z_k \geqslant z_\ell \quad \text{for every } k \in \mathcal{I}_i \text{ and } \ell \in \mathcal{I}_{i+1} \text{ with } i = 1, 2, \ldots, K - 1. \tag{18}$$

For one fixed (i, ℓ) pair, (18) gives $|\mathcal{I}_i| = m_i$ inequalities, one for each $k \in \mathcal{I}_i$. These inequalities can be gathered into an elementwise matrix inequality as

$$[0_{m_i \times M_{i-1}} \quad I_{m_i} \quad 0_{m_i \times (M-M_i)}]z \geqslant [0_{m_i \times (\ell-1)} \quad 1_{m_i \times 1} \quad 0_{m_i \times (M-\ell)}]z,$$

where $M_k = m_1 + m_2 + \cdots + m_k$, or $D_{i,\ell}^{(m)} z \geqslant 0_{m_i \times 1}$, where

$$D_{i,\ell}^{(m)} = [0_{m_i \times M_{i-1}} \quad I_{m_i} \quad 0_{m_i \times (\ell - M_i - 1)} \quad -1_{m_i \times 1} \quad 0_{m_i \times M - \ell}] \tag{19a}$$

is an $m_i \times M$ differencing matrix. Allowing ℓ to vary across $\mathcal{I}_{i+1}$, we define the $m_i m_{i+1} \times M$ matrix

$$D_i^{(m)} = \begin{bmatrix} D_{i,M_i+1}^{(m)} \\ D_{i,M_i+2}^{(m)} \\ \vdots \\ D_{i,M_i+m_{i+1}}^{(m)} \end{bmatrix} \tag{19b}$$

and express all of (18) for one fixed i as $D_i^{(m)} z \geqslant 0_{m_i m_{i+1} \times 1}$.

Continuing our recursion, it only remains to gather the inequalities (18) across $i \in \{1, 2, \ldots, K - 1\}$. Let

$$D^{(m)} = \begin{bmatrix} D_1^{(m)} \\ D_2^{(m)} \\ \vdots \\ D_{K-1}^{(m)} \end{bmatrix}, \tag{19c}$$

which has

$$L(m) = \sum_{i=1}^{K-1} m_i m_{i+1} \tag{20}$$

rows. The property of z being m-descending can be expressed as $D^{(m)} z \geqslant 0_{L(m) \times M}$. The following example illustrates the form of $D^{(m)}$:

$$D^{((2,3,2))} = \begin{bmatrix} 1 & 0 & -1 & 0 & 0 & 0 & 0 \\ 0 & 1 & -1 & 0 & 0 & 0 & 0 \\ 1 & 0 & 0 & -1 & 0 & 0 & 0 \\ 0 & 1 & 0 & -1 & 0 & 0 & 0 \\ 1 & 0 & 0 & 0 & -1 & 0 & 0 \\ 0 & 1 & 0 & 0 & -1 & 0 & 0 \\ 0 & 0 & 1 & 0 & 0 & -1 & 0 \\ 0 & 0 & 0 & 1 & 0 & -1 & 0 \\ 0 & 0 & 0 & 0 & 1 & -1 & 0 \\ 0 & 0 & 1 & 0 & 0 & 0 & -1 \\ 0 & 0 & 0 & 1 & 0 & 0 & -1 \\ 0 & 0 & 0 & 0 & 1 & 0 & -1 \end{bmatrix}.$$

Notice the important property that each row of $D^{(m)}$ has one 1 entry and one -1 entry with the remaining entries 0. This will be exploited in Section 5.

Now we can apply these representations to FPQ.

Variant I: In this case, we know PFx is m-descending. Consistency is thus simply expressed as

$$D^{(m)} PFx \geqslant 0. \tag{21}$$

Variant II: The second variant has an m-descending property after V has made the signs of the significant frame coefficients (all but last m_K) positive: $D^{(m)} VPFx \geqslant 0$. In addition, we have the nonnegativity of all of the first $M - m_K$ sorted and sign-changed coefficients. To specify

$$[I_{M-m_K} \quad 0_{(M-m_K) \times M}]VPFx \geqslant 0_{(M-m_K) \times 1}$$

Algorithm 3: Estimation of source on $[-\frac{1}{2}, \frac{1}{2}]^N$ for Variant I frame permutation quantization.

Inputs:	Analysis frame operator F, composition m, and FPQ encoding P
Output:	Estimate $\hat{x}$ consistent with (F, m, P) and as far from the partition boundaries as possible

     1. Let $A = \begin{bmatrix} -D^{(m)}PF & 1_{L(m)\times 1} \\ -I_N & 1_{N\times 1} \\ I_N & 1_{N\times 1} \end{bmatrix}$ and $b = \frac{1}{2}\begin{bmatrix} 0_{L(m)\times 1} \\ 1_{2N\times 1} \end{bmatrix}$,

        where $D^{(m)}$ is defined in (19) and $L(m)$ is defined in (20).
        (Consistency with (21) and $x \in [-\frac{1}{2}, \frac{1}{2}]^N$ is expressed as $A\begin{bmatrix} x \\ 0 \end{bmatrix} \leqslant b$.)
     2. Let $c = \begin{bmatrix} 0_{N\times 1} \\ -1 \end{bmatrix}$.
     3. Use a linear programming method to minimize $c^T\begin{bmatrix} x \\ \delta \end{bmatrix}$ subject to $A\begin{bmatrix} x \\ \delta \end{bmatrix} \leqslant b$.
        Return the first N components of the minimizer as $\hat{x}$.

---

is redundant with what is expressed with the m-descending property. The added constraints can be applied only to the entries of $VPFx$ with indexes in $\mathcal{I}_{K-1}$ because all the earlier entries are already ensured to be larger. We thus express consistency as

$$\underbrace{\begin{bmatrix} & D^{(m)} & \\ 0_{m_{K-1}\times M_{K-1}} & I_{m_{K-1}} & 0_{m_{K-1}\times m_K} \end{bmatrix}}_{\widetilde{D}^{(m)}} VPFx \geqslant 0. \tag{22}$$

4.3. Consistent reconstruction algorithms

The constraints (21) and (22) both specify unbounded sets, as discussed previously and illustrated in Fig. 1(e) and (f). To be able to decode FPQs in analogy to Algorithm 1, we require some additional constraints. We develop two examples: a source x bounded to $[-\frac{1}{2}, \frac{1}{2}]^N$ (e.g., having an i.i.d. uniform distribution over $[-\frac{1}{2}, \frac{1}{2}]$) or having an i.i.d. standard Gaussian distribution. For the remainder of this section, we consider only Variant I because adjusting for Variant II using (22) is easy.

4.3.1. Source bounded to $[-\frac{1}{2}, \frac{1}{2}]^N$
To impose (21) along with $x \in [-\frac{1}{2}, \frac{1}{2}]^N$ is trivial because $x \in [-\frac{1}{2}, \frac{1}{2}]^N$ is decomposable into $2N$ inequality constraints:

$$\begin{bmatrix} I_N \\ -I_N \end{bmatrix} x \leqslant \frac{1}{2}\begin{bmatrix} 1_{N\times 1} \\ -1_{N\times 1} \end{bmatrix}.$$

A linear programming formulation will return some corner of the consistent set, depending on the choice of cost function. The unknown vector x can be augmented with a variable δ that represents the slackness of the inequality constraint with the least slack. Maximizing δ moves the solution away from the boundary of the consistent set (partition cell) as much as possible. Reconstruction using this principle is outlined in Algorithm 3.

If the source x is random and the distribution $p(x)$ is known, then one could optimize some criterion explicitly. For example, one could maximize $p(x)$ over the consistent set or compute the centroid of the consistent set with respect to $p(x)$. This would improve upon reconstructions computed with Algorithm 3 but presumably increase complexity greatly.

4.3.2. Source with i.i.d. standard Gaussian distribution
Suppose x has i.i.d. Gaussian components with mean zero and unit variance. Since the source support is unbounded, something beyond consistency must be used in reconstruction. Here we use a quadratic program to find a good bounded, consistent estimate and combine this with the average value of $\|x\|$.

The problem with using (21) combined with maximization of minimum slackness alone (without any additional boundedness constraints) is that for any purported solution, multiplying by a scalar larger than 1 will increase the slackness of all the constraints. Thus, any solution technique will naturally and correctly have $\|\hat{x}\| \to \infty$. Actually, because the partition cells are convex cones, we should not hope to recover the radial component of x from the partition. Instead, we should only hope to recover a good estimate of $x/\|x\|$.

To estimate the angular component $x/\|x\|$ from the partition, it would be convenient to maximize minimum slackness while also imposing a constraint of $\|\hat{x}\| = 1$. Unfortunately, this is a nonconvex constraint. It can be replaced by $\|\hat{x}\| \leqslant 1$ because slackness is proportional to $\|\hat{x}\|$. This suggests the optimization

     maximize δ subject to $\|x\| \leqslant 1$ and $D^{(m)}PFx \geqslant \delta 1_{L(m)\times 1}$.

Denoting the x at the optimum by $\hat{x}_{\text{ang}}$, we still need to choose the radial component, or length, of $\hat{x}$.

For the $\mathcal{N}(0, I_N)$ source, the mean length is [45]

$$E\big[\|x\|\big] = \frac{\sqrt{2\pi}}{\beta(N/2, 1/2)} \approx \sqrt{N - 1/2}.$$

We can combine this with $\hat{x}_{\text{ang}}$ to obtain a reconstruction $\hat{x}$.

Algorithm 4: Estimation of $\mathcal{N}(0, I_N)$ source for Variant I frame permutation quantization.

Inputs:	Analysis frame operator F, composition m, and FPQ encoding P
Output:	Estimate $\hat{x}$ consistent with (F, m, P) and as far from the partition boundaries as possible
	while keeping $\|\hat{x}\| = E[\|x\|]$

    1. Let $A = [-D^{(m)} PF \quad 1_{L(m) \times 1}]$ and $b = 0_{L(m) \times 1}$,
        where $D^{(m)}$ is defined in (19) and $L(m)$ is defined in (20).
        (Consistency with (21) is expressed as $A \begin{bmatrix} x \\ 0 \end{bmatrix} \leqslant b$.)
    2. Let $c = \begin{bmatrix} 0_{N \times 1} \\ -1 \end{bmatrix}$ and $H = \begin{bmatrix} I_N & 0_{N \times 1} \\ 0_{1 \times N} & 0 \end{bmatrix}$.
    3. Use a quadratic programming method to minimize $\frac{1}{2} \begin{bmatrix} x \\ \delta \end{bmatrix}^T H \begin{bmatrix} x \\ \delta \end{bmatrix} + c^T \begin{bmatrix} x \\ \delta \end{bmatrix}$
        subject to $A \begin{bmatrix} x \\ \delta \end{bmatrix} \leqslant b$. Denote the first N components of the minimizer as $\hat{x}_{\text{ang}}$.
    4. Return $(\sqrt{2\pi} / \beta(N/2, 1/2)) \hat{x}_{\text{ang}}$.

We use a slightly different formulation to have a quadratic program in standard form. We combine the radial component constraint with the goal of maximizing slackness to obtain

$$\text{minimize } \frac{1}{2} x^T x - \lambda \delta \text{ subject to } - D^{(m)} PF x \leqslant -\delta 1_{L(m) \times 1},$$

where λ trades off slackness against the radial component of x. Since the radial component will be replaced with its expectation, the choice of λ is immaterial; it is set to 1 in Algorithm 4.

4.4. Recursive estimation algorithms

Algorithms 3 and 4 are practical for small values of N and M, as one would encounter in data compression applications, but not for large values of N and M, as may be of interest in data acquisition. Specifically, Algorithm 3 uses a linear program with $N + 1$ variables and $L(m) + 2N$ constraints, while Algorithm 4 uses a quadratic program with $N + 1$ variables and $L(m)$ constraints. The costs of these computations are superlinear in $L(m)$, and $L(m)$ is at least $M - 1$.

One way to lower the reconstruction complexity is to sacrifice global consistency in favor of recursive computability, in analogy to Algorithm 2. For recursive algorithms, we restrict our attention to the composition $m = (1, 1, \ldots, 1)$. We also again restrict our attention to Variant I because adjusting for Variant II is straightforward.

With all-1s compositions, FPQ encoding produces a *successive* or *embedded* representation of x: a representation with a k-element analysis frame is a ranking of $\{\langle x, \phi_j \rangle\}_{j=1}^k$ (for Variant I), and adding a vector ϕ_{k+1} to the analysis frame amounts to inserting $\langle x, \phi_{k+1} \rangle$ in the ranked list. Equivalently, the encoding of x with a k-element frame is the set

$$\text{sign}(\langle x, \phi_i \rangle - \langle x, \phi_j \rangle), \quad i, j \in \{1, 2, \ldots, k\},$$

and adding ϕ_{k+1} to the analysis frame adds

$$\text{sign}(\langle x, \phi_{k+1} \rangle - \langle x, \phi_j \rangle), \quad j \in \{1, 2, \ldots, k\},$$

to the representation without removing any of the previous information.

In estimating x from FPQ with the full $(k + 1)$-element analysis frame, one could impose

$$\text{sign}(\langle \hat{x}, \phi_i - \phi_j \rangle) = \text{sign}(\langle x, \phi_i - \phi_j \rangle), \quad i, j \in \{1, 2, \ldots, k + 1\} \tag{23}$$

(equivalent to (21)) for *global* consistency. However, for a recursive computation we will compute an estimate $\hat{x}_{k+1}$ from an estimate $\hat{x}_k$ and some subset of the constraints

$$\text{sign}(\langle \hat{x}, \phi_{k+1} - \phi_j \rangle) = \text{sign}(\langle x, \phi_{k+1} - \phi_j \rangle), \quad j \in \{1, 2, \ldots, k\}. \tag{24}$$

Updating $\hat{x}_k$ to satisfy any of the constraints (24) can cause constraints (23) with $i < k + 1$ to be violated, so a strategy of imposing *local* consistency does not ensure global consistency. However, we will demonstrate by extending results from [8, 12] that optimal MSE decay as a function of M can still be obtained.

A recursive computation that uses local consistency is described explicitly in Algorithm 5. For each k, the set $\mathcal{J}_k$ represents the indexes j for which the constraint (24) is used. Any one of them is used (in Step 3b) by orthogonally projecting the running estimate $\hat{x}_{k-1}$ to the half-space consistent with (24). This yields a monotonicity result analogous to [8, Thm. 1] and [12, Lem. 7.2]:

Theorem 4.2. *Let $x \in \mathbb{R}^N$ be a unit vector. The sequence of estimates produced in Algorithm 5 satisfies*

$$\|x - \hat{x}_{k+1}\| \leqslant \|x - \hat{x}_k\|.$$

Algorithm 5: Recursive estimation for Variant I frame permutation quantization.

Inputs:	Analysis frame sequence $\{\phi_j\}_{j=1}^M$, FPQ representation $\nu_{i,j} = \text{sign}(\langle x, \phi_i - \phi_j\rangle)$ for $j = 1, 2, \ldots, M$, $i = 1, 2, \ldots, j - 1$, and index sets $\mathcal{J}_2, \mathcal{J}_3, \ldots, \mathcal{J}_M$ with $\mathcal{J}_k \subset \{1, 2, \ldots, k - 1\}$
Output:	Estimate $\hat{x}_M$

    1. Let $\hat{x}_1$ be a vector chosen uniformly at random from the unit sphere in $\mathbb{R}^N$ and let $k = 2$.
    2. Let $\hat{x}_k = \hat{x}_{k-1}$.
    3. For each $j \in \mathcal{J}_k$, taken in random order:
        3a. Let $\psi_{k,j} = \phi_k - \phi_j$.
        3b. If $\text{sign}(\langle \hat{x}_k, \psi_{k,j}\rangle) \neq \nu_{k,j}$, update $\hat{x}_k$ to $\hat{x}_k - \langle \hat{x}_k, \psi_{k,j}\rangle \psi_{k,j}/\|\psi_{k,j}\|^2$.
    4. If $k = M$, return $\hat{x}_k/\|\hat{x}_k\|$; else increment k and go to step 2.

Proof. Since Step 3b is an orthogonal projection to a convex set containing x, no occurrence of this step can increase the estimation error. □

The number of projection steps in Algorithm 5 depends on the sizes of the $\mathcal{J}_k$s. At one extreme, each $\mathcal{J}_k$ is a singleton so there are $M - 1$ projections. At the other extreme, each $\mathcal{J}_k$ has $k - 1$ elements and there are $\frac{1}{2}M(M-1)$ projections. The empirical behavior in Section 6.3 shows that the MSE decays as the square of the number of projection steps. This behavior is provable in some cases. One such result is the following theorem, analogous to [8, Thm. 2] and presumably extendable to match [12, Thm. 7.9]:

Theorem 4.3. *Let x and $\{\phi_k\}_{k=1}^\infty$ be i.i.d. vectors drawn from the uniform distribution on the unit sphere in $\mathbb{R}^N$, and for each $k \in \{2, 3, \ldots\}$, let $\mathcal{J}_k = \{1\}$. Then the normalized sequence of estimates produced by Algorithm 5 satisfies*

$$\|x - \hat{x}_k\|^2 k^p \to 0 \quad \textit{almost surely, for every } p < 2. \tag{25}$$

Proof. We give only a brief sketch of a proof since the main ideas have been developed by Rangan and Goyal [8] and Powell [12].

According to [8, Thm. 2], Algorithm 2 gives performance satisfying (25) under the following assumptions:

1. Quantization noise is on a known, bounded interval;
2. the frame sequence is i.i.d. and independent of the quantization noise; and
3. the frame vectors are bounded and satisfy

$$E\big[\big|\langle z, \phi_k\rangle\big|\big] \geqslant \varepsilon \|z\| \quad \text{for all } z \in \mathbb{R}^N$$

for some $\varepsilon > 0$.

Assumption 1 can be loosened to quantization noise known to lie in $[-1, \infty)$ or $(-\infty, 1]$ without changing the conclusion that (25) holds; it is qualitatively like discarding half of the quantized frame coefficients since each frame coefficient gives one half-space constraint rather than two. Assumption 3 is a very mild condition that simply ensures that there is no nonzero vector $z \in \mathbb{R}^N$ such that all of the probability mass of the frame vector distribution is orthogonal to z; such orthogonality implies $\{\langle x, \phi_k\rangle\}_{k=1}^\infty$ gives no information about the component of x in the subspace generated by z and hence makes recovery from the inner product sequence impossible even without quantization.

An FPQ representation is through $\text{sign}(\langle x, \psi_{k,j}\rangle)$ where $\psi_{k,j} = \phi_k - \phi_j$, $j \in \mathcal{J}_k$, and $k = 2, 3, \ldots$. This is equivalent to a quantized frame expansion with analysis vectors $\psi_{k,j}$ and quantization noise bounded to $[-1, \infty)$ when the signum function returns 1 and bounded to $(-\infty, 1]$ when the signum function returns -1. Due to the symmetric distribution of x and its independence of the frame sequence, the quantization noise is independent of the frame sequence.

We are considering reconstruction where each $\mathcal{J}_k$ is $\{1\}$, so we are interested in whether quantized versions of $\{\langle x, \psi_{k,1}\rangle\}$ are adequate to ensure the error decay (25). While the vectors $\{\psi_{k,1}\}_{k=2}^\infty$ are identically distributed and (through spherical symmetry) satisfy Assumption 3 above, they are not independent. Nevertheless, we can employ [8, Thm. 2] as follows: Conditioned on any value of ϕ_1, the vectors $\{\psi_{k,1}\}_{k=2}^\infty$ are conditionally independent, so by eliminating the radial component of x, one can conclude that the error decay (25) holds conditionally. Since the almost sure convergence in (25) holds under every conditional probability law specified by ϕ_1, it must also hold unconditionally. □

Extensions of [8, Thm. 2] and [12, Thm. 7.9] to non-i.i.d. frame sequences would lead to extensions of Theorem 4.3 beyond singleton $\mathcal{J}_k$s.

5. Conditions on the choice of frame

In this section, we provide necessary and sufficient conditions so that a linear reconstruction is also consistent. We first consider a general linear reconstruction, $\hat{x} = R\hat{y}$, where R is some $N \times M$ matrix and $\hat{y}$ is a decoding of the PSC of y. We

then restrict attention to canonical reconstruction, where $R = F^\dagger$. For each case, we describe all possible choices of a "good" frame F, in the sense of the consistency of the linear reconstruction.

5.1. Arbitrary linear reconstruction

We begin by introducing a useful term.

Definition 5.1. A matrix is called *column-constant* when each column of the matrix is a constant. The set of all $M \times M$ column-constant matrices is denoted $\mathcal{J}$.

We now give our main results for arbitrary linear reconstruction combined with FPQ decoding of an estimate of y.

Theorem 5.2. *Suppose $A = FR = aI_M + J$ for some $a \geqslant 0$ and $J \in \mathcal{J}$. Then the linear reconstruction $\hat{x} = R\hat{y}$ is consistent with Variant* I *FPQ encoding using frame F, an arbitrary composition and an arbitrary Variant* I *initial codeword compatible with it.*

Proof. We start the proof by pointing out two special properties of any matrix $J \in \mathcal{J}$:

(P1) $PJP^{-1} \in \mathcal{J}$ for any permutation matrix P; and

(P2) $D^{(m)}J = 0_{L(m) \times 1}$ for any composition m.

(P1) follows from the fact that neither left multiplying by P nor right multiplying by P^{-1} disturbs column-constancy. (P2) is true because each row of $D^{(m)}$ has zero entries except for one 1 and one -1.

Suppose $m = (m_1, m_2, \ldots, m_K)$ is an arbitrary composition of M and $\hat{y}_{\text{init}}$ is an arbitrary Variant I initial codeword compatible with m. Let P be the Variant I FPQ encoding of x using $(F, m, \hat{y}_{\text{init}})$. We would like to check that $\hat{x} = R\hat{y}$ is consistent with the encoding P. This is verified through the following computation:

$$
\begin{aligned}
D^{(m)}PF\hat{x} &= D^{(m)}PFR\hat{y} \\
&= D^{(m)}PFRP^{-1}\hat{y}_{\text{init}} \quad\quad\quad\quad\quad\quad\quad\quad\quad\quad\quad (26) \\
&= D^{(m)}PAP^{-1}\hat{y}_{\text{init}} \\
&= D^{(m)}P(aI_M + J)P^{-1}\hat{y}_{\text{init}} \quad\quad\quad\quad\quad\quad\quad\quad (27) \\
&= aD^{(m)}\hat{y}_{\text{init}} + D^{(m)}\hat{J}\hat{y}_{\text{init}} \quad \text{for some } \hat{J} \in \mathcal{J} \quad\quad (28) \\
&= aD^{(m)}\hat{y}_{\text{init}} \quad\quad\quad\quad\quad\quad\quad\quad\quad\quad\quad\quad\quad (29) \\
&\geqslant 0_{L(m) \times 1}, \quad\quad\quad\quad\quad\quad\quad\quad\quad\quad\quad\quad\quad\quad\quad (30)
\end{aligned}
$$

where (26) uses the conventional decoding of a PSC; (27) follows from the hypothesis of the theorem on A; (28) follows from (P1); (29) follows from (P2); and (30) follows from the definition of Variant I initial codewords compatible with m, and the nonnegativity of a. This completes the proof. $\square$

The key point of the proof of Theorem 5.2 is showing that the inequality

$$
D^{(m)}PAP^{-1}\hat{y}_{\text{init}} \geqslant 0, \quad\quad\quad\quad\quad\quad\quad\quad\quad\quad\quad\quad\quad\quad\quad (31)
$$

where $A = FR$, holds for every composition m and every initial codeword $\hat{y}_{\text{init}}$ compatible with it. It turns out that the form of matrix A given in Theorem 5.2 is the unique form that guarantees that (31) holds for every pair $(m, \hat{y}_{\text{init}})$. In other words, the condition on A that is sufficient for any composition m and any initial codeword $\hat{y}_{\text{init}}$ compatible with it is also necessary for consistency for every pair $(m, \hat{y}_{\text{init}})$.

Theorem 5.3. *Consider Variant* I *FPQ using frame F with $M \geqslant 3$. If linear reconstruction $\hat{x} = R\hat{y}$ is consistent with every composition and every Variant* I *initial codeword compatible with it, then matrix $A = FR$ must be of the form $aI_M + J$, where $a \geqslant 0$ and $J \in \mathcal{J}$.*

Proof. See Section 7.2. $\square$

The column-constant matrices are those obtained by multiplying an all-1s matrix on the right by a diagonal matrix. Thus, except in the case of an all-0s matrix, a column-constant matrix has rank 1. A matrix of the form $aI_M + J$ where $a > 0$ and $J \in \mathcal{J}$ thus has rank $M - 1$ or M. Since $A = FR$ has rank at most N because of the dimensions of F and R, the necessary and sufficient condition from Theorems 5.2 and 5.3 imply $M = N$ or $M = N + 1$.

Similar necessary and sufficient conditions can be derived for linear reconstruction of Variant II FPQs. Since the partition cell associated with a codeword of a Variant II FPQ is much smaller than that of the corresponding Variant I FPQ, we expect

the condition for a linear reconstruction to be consistent to be more restrictive than that given in Theorems 5.2 and 5.3. The following two theorems show that this is indeed the case.

Theorem 5.4. *Suppose $A = FR = aI_M$ for some $a \geqslant 0$ and $M = N$. Then the linear reconstruction $\hat{x} = R\hat{y}$ is consistent with Variant II FPQ encoding using frame F, an arbitrary composition, and an arbitrary Variant II initial codeword compatible with it.*

Proof. Suppose that $m = (m_1, m_2, \ldots, m_K)$ is an arbitrary composition of M and $\hat{y}_{\text{init}}$ is an arbitrary Variant II initial codeword compatible with it. Let (P, V) be the Variant II FPQ encoding of x using $(F, m, \hat{y}_{\text{init}})$. We would like to check that $\hat{x} = R\hat{y}$ is consistent with the encoding (P, V). This is verified through the following computation:

$$\widetilde{D}^{(m)} V P F \hat{x} = \widetilde{D}^{(m)} V P F R \hat{y}$$
$$= \widetilde{D}^{(m)} V P F R P^{-1} V^{-1} \hat{y}_{\text{init}} \tag{32}$$
$$= \widetilde{D}^{(m)} V P A P^{-1} V^{-1} \hat{y}_{\text{init}}$$
$$= \widetilde{D}^{(m)} V P a I_M P^{-1} V^{-1} \hat{y}_{\text{init}} \tag{33}$$
$$= a \widetilde{D}^{(m)} \hat{y}_{\text{init}}$$
$$\geqslant 0_{L(m) \times 1}, \tag{34}$$

where (32) uses the conventional decoding of a PSC; (33) follows from the hypothesis of the theorem on A; and (34) follows from the definition of Variant II initial codewords compatible with m, and the nonnegativity of a. This completes the proof. □

Theorem 5.5. *Consider Variant II FPQ using frame F with $M \geqslant 3$. If linear reconstruction $\hat{x} = R\hat{y}$ is consistent with every composition and every Variant II initial codeword compatible with it, then matrix $A = FR$ must be of the form aI_M, where $a \geqslant 0$ and $M = N$.*

Proof. See Section 7.3. □

The two theorems above show that, if we insist on linear consistent reconstructions for Variant II FPQs, the frame must degenerate into a basis. For nonlinear consistent reconstructions, we could use algorithms analogous to those presented in Section 4.3 for an arbitrary frame that is not necessarily a basis.

5.2. Canonical reconstruction

We now restrict the linear reconstruction to use the canonical dual; i.e., R is restricted to be the pseudo-inverse $F^{\dagger} = (F^* F)^{-1} F^*$. The following corollary characterizes the nontrivial frames for which canonical reconstructions are consistent.

Corollary 5.6. *Consider Variant I FPQ using rank-N frame F with $M > N$ and $M \geqslant 3$. For canonical reconstruction to be consistent with every composition and every Variant I initial codeword compatible with it, it is necessary and sufficient to have $M = N + 1$ and $A = FF^{\dagger} = I_M - \frac{1}{M} J_M$, where J_M is the $M \times M$ all-1s matrix.*

Proof. Sufficiency follows immediately from Theorem 5.2. From Theorem 5.3, it is necessary to have $A = FF^{\dagger} = aI_M + J$ for some $a \geqslant 0$ and $J \in \mathcal{J}$. The rank condition further implies $a > 0$, so we must have $M = N + 1$ by the argument following Theorem 5.3. Now since A is an orthogonal projection operator, it is self-adjoint so

$$aI_M + J = (aI_M + J)^* = aI_M + J^*. \tag{35}$$

Thus, $J = J^*$, and it follows that $J = bJ_M$, for some constant b. The idempotence of A gives

$$aI_M + bJ_M = (aI_M + bJ_M)^2 = a^2 I_M + (2ab + b^2 M) J_M. \tag{36}$$

Equating the two sides of (36) yields $a = 1$ and $b = -1/M$ as desired. □

We continue to add more constraints to frame F. Tightness and equal norm are common requirements in frame design [1]. By imposing tightness and unit norm on our analysis frame, we can progress a bit further from Corollary 5.6 to derive the form of FF^*.

Corollary 5.7. *Consider Variant I FPQ using unit-norm tight frame F with $M > N$ and $M \geqslant 3$. For canonical reconstruction to be consistent for every composition and every Variant I initial codeword compatible with it, it is necessary and sufficient to have $M = N + 1$ and*

$$
FF^* = \begin{bmatrix} 1 & -\frac{1}{N} & \cdots & -\frac{1}{N} \\ -\frac{1}{N} & 1 & \cdots & -\frac{1}{N} \\ \vdots & \vdots & \ddots & \vdots \\ -\frac{1}{N} & -\frac{1}{N} & \cdots & 1 \end{bmatrix}.
\tag{37}
$$

Proof. Corollary 5.6 asserts that $M = N + 1$ and

$$
F(F^*F)^{-1}F^* = \begin{bmatrix} \frac{N}{M} & -\frac{1}{M} & \cdots & -\frac{1}{M} \\ -\frac{1}{M} & \frac{N}{M} & \cdots & -\frac{1}{M} \\ \vdots & \vdots & \ddots & \vdots \\ -\frac{1}{M} & -\frac{1}{M} & \cdots & \frac{N}{M} \end{bmatrix}.
\tag{38}
$$

On the other hand, the tightness of a unit-norm frame F implies

$$
(F^*F)^{-1} = \frac{N}{M}I_N.
\tag{39}
$$

Combining (38) with (39), we get (37). $\quad\square$

Recall that a UNTF that satisfies (37) is a restricted ETF. Therefore Corollary 5.7 together with Proposition 3.7 gives us a complete characterization of UNTFs that are "good" in the sense of canonical reconstruction being consistent.

Corollary 5.8. *Consider Variant* I *FPQ using unit-norm tight frame F with $M > N$ and $M \geqslant 3$. For canonical reconstruction to be consistent for every composition and every Variant* I *initial codeword compatible with it, it is necessary and sufficient for F to be a modulated HTF or a Type* I *or Type* II *equivalent.*

6. Numerical results

In this section, we provide simulations to demonstrate some properties of FPQ. For data compression, we demonstrate that FPQ with decoding using Algorithms 3 and 4 can give performance better than entropy-constrained scalar quantization (ECSQ) and ordinary PSC for certain combinations of signal dimension and rate. For data acquisition, we demonstrate that FPQ with recursive estimation through Algorithm 5 empirically gives the optimal decay of MSE, inversely proportional to the square of the number of orthogonal projection steps, validating Theorem 4.3 but also suggesting that this holds more generally.

6.1. Fixed-rate compression experiments

All FPQ compression simulations use modulated harmonic tight frames and are based on implementations of Algorithms 3 and 4 using MATLAB, with linear programming and quadratic programming provided by the Optimization Toolbox. For every data point shown, the distortion represents a sample mean estimate of $N^{-1}E[\|x - \hat{x}\|^2]$ over at least 10^6 trials. Testing was done with exhaustive enumeration of the relevant compositions. This makes the complexity of simulation high, and thus experiments are only shown for small N and M. Recall the encoding complexity of FPQ is low, $O(M \log M)$ operations. The decoding complexity is polynomial in M for either of the algorithms presented explicitly, and in some applications it could be worthwhile to precompute the entire codebook at the decoder. Thus much larger values of N and M than used here may be practical.

6.1.1. Uniform source

Let x have i.i.d. components uniformly distributed on $[-\frac{1}{2}, \frac{1}{2}]$. Algorithm 3 is clearly well suited to this source since the support of x is properly specified and reconstructions near the centers of cells is nearly optimal. Fig. 2 summarizes the performance of Variant I FPQ for several frames and an enormous number of compositions. Also shown are the performances of ordinary PSC and optimal ECSQ [32,46]. Approaching the indicated performance of ECSQ requires entropy coding over many quantized symbols.

Using $F = I_N$ makes FPQ reduce to ordinary PSC. We see that, consistent with results in [32], PSC is sometimes better than ECSQ. Next notice that FPQ is not identical to PSC when F is square but not the identity matrix. The modulated harmonic frame with $M = N$ provides an orthogonal matrix F. The set of rates obtained with $M = N$ is the same as PSC, but since the source is not rotationally-invariant, the partitions and hence distortions are not the same; the distortion is sometimes better and sometime worse. Increasing M gives more operating points—some of which are better than those for
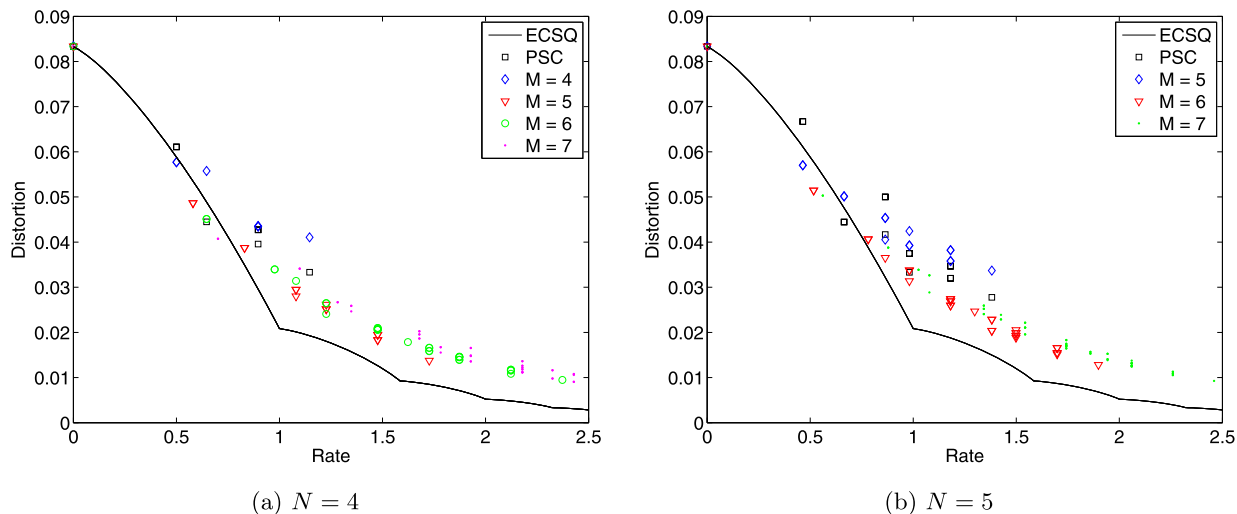
(a) $N = 4$

(b) $N = 5$

Fig. 2. Performance of Variant I FPQ on an i.i.d. uniform($[-\frac{1}{2}, \frac{1}{2}]$) source using modulated harmonic tight frames ranging in size from N to 7. Also shown are the performances of ordinary PSC (equivalent to FPQ with frame $F = I_N$), and optimal entropy-constrained scalar quantization.
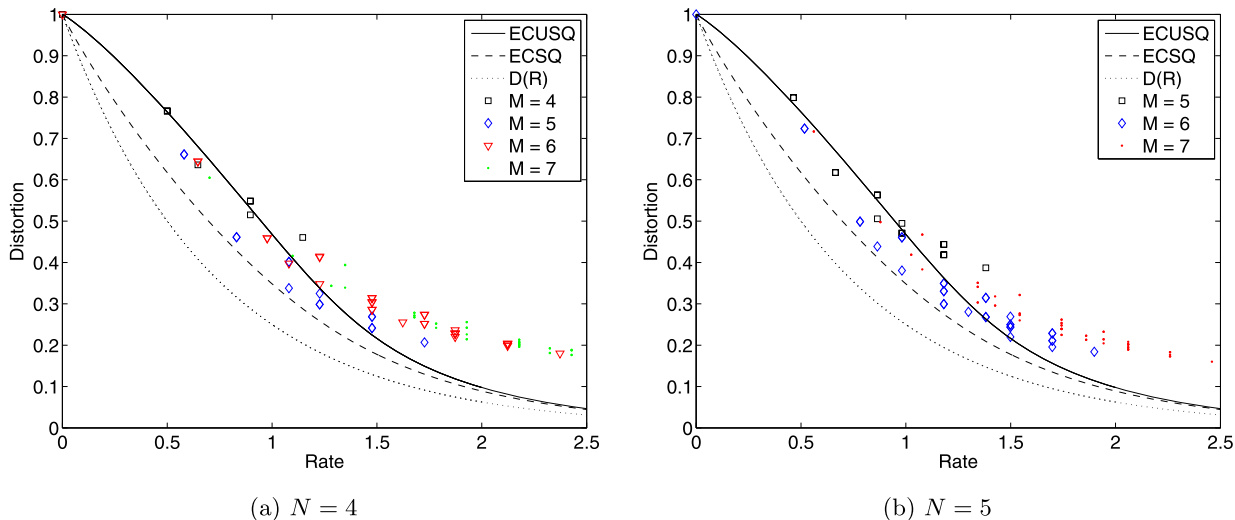

(a) $N = 4$

(b) $N = 5$

Fig. 3. Performance of Variant I FPQ on an i.i.d. $\mathcal{N}(0, 1)$ source using modulated harmonic tight frames ranging in size from N to 7. Performance of PSC is not shown because it is equivalent to FPQ with $M = N$ for this source. Also plotted are the performance of entropy-constrained scalar quantization and the distortion–rate bound.

lower M—and a higher maximum rate.[4] In particular, for both $N = 4$ and $N = 5$, it seems that $M = N + 1$ gives several operating points better than those obtainable with larger or smaller values of M.

6.1.2. Gaussian source

Let x have the $\mathcal{N}(0, I_N)$ distribution. Algorithm 4 is designed precisely for this source. Fig. 3 summarizes the performance of Variant I FPQ with decoding using Algorithm 4. Also shown are the distortion–rate bound and the performances of two types of entropy-constrained scalar quantization: uniform thresholds with uniform codewords (labeled ECUSQ) and uniform thresholds with optimal codewords (labeled ECSQ). At all rates, the latter is a very close approximation to optimal ECSQ; in particular, it has optimal rate–distortion slope at rate zero [48]. Of course, the distortion–rate bound can only be approached with $N \to \infty$; it is not presented as a competitive alternative to FPQ for $N = 4$ and $N = 5$.

We have not provided an explicit comparison to ordinary PSC because, due to rotational-invariance of the Gaussian source, FPQ with any orthonormal basis as the frame is identical to PSC. (The modulated harmonic tight frame with $M = N$

---

[4] A discussion of the density of PSC rates is given in [47, App. B].
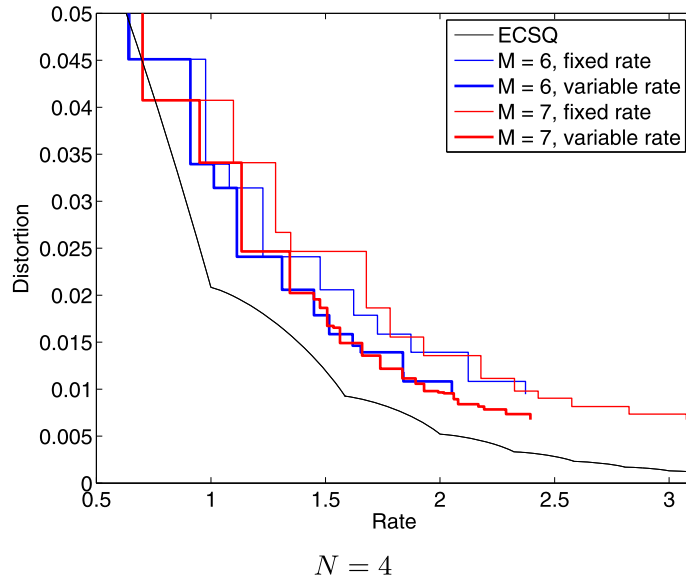
$$N = 4$$

Fig. 4. Performance of Variant I FPQ for fixed- and variable-rate coding of an i.i.d. uniform $([-\frac{1}{2}, \frac{1}{2}])$ source with $N = 4$ using modulated harmonic tight frames of sizes 6 and 7. Also plotted is the performance of entropy-constrained scalar quantization.

is an orthonormal basis.) The trends are similar to those for the uniform source: PSC and FPQ are sometimes better than ECSQ; increasing M gives more operating points and a higher maximum rate; and $M = N + 1$ seems especially attractive.

6.2. Variable-rate compression experiments and discussion

We have posed FPQ as a fixed-rate coding technique. As mentioned in Section 3.2, symmetries will often make the outputs of a PSC equally likely, making variable-rate coding superfluous. This does not necessarily carry over to FPQ.

In Variant I FPQ with modulated HTFs, when $M > N + 1$ the codewords are not only nonequiprobable, some cannot even occur. To see an example of this, consider the case of $(N, M) = (2, 4)$. Then

$$F = \begin{bmatrix} 1 & 0 \\ -\zeta & -\zeta \\ 0 & 1 \\ \zeta & -\zeta \end{bmatrix}, \quad \text{where } \zeta \text{ denotes } 1/\sqrt{2}.$$

If we choose the composition $m = (2, 2)$, we might expect six distinct codewords that are equiprobable for a rotationally-invariant source. The permutation matrices consistent with this composition are

$$\left\{ \begin{bmatrix} 0&0&1&0 \\ 0&0&0&1 \\ 1&0&0&0 \\ 0&1&0&0 \end{bmatrix}, \begin{bmatrix} 0&0&1&0 \\ 1&0&0&0 \\ 0&1&0&0 \\ 0&0&0&1 \end{bmatrix}, \begin{bmatrix} 0&0&1&0 \\ 1&0&0&0 \\ 0&0&0&1 \\ 0&1&0&0 \end{bmatrix}, \begin{bmatrix} 1&0&0&0 \\ 0&0&1&0 \\ 0&1&0&0 \\ 0&0&0&1 \end{bmatrix}, \begin{bmatrix} 1&0&0&0 \\ 0&1&0&0 \\ 0&0&1&0 \\ 0&0&0&1 \end{bmatrix}, \begin{bmatrix} 1&0&0&0 \\ 0&0&1&0 \\ 0&0&0&1 \\ 0&1&0&0 \end{bmatrix} \right\}.$$

The first and fifth of these occur with probability zero because the corresponding partition cells have zero volume. Let us verify this for the fifth permutation matrix ($P = I_4$). By forming $D^{((2,2))} I_4 F$, we see that the fifth cell is described by

$$\begin{bmatrix} 1 & -1 \\ -\zeta & -1-\zeta \\ 1-\zeta & \zeta \\ -2\zeta & 0 \end{bmatrix} \begin{bmatrix} x_1 \\ x_2 \end{bmatrix} \geqslant \begin{bmatrix} 0 \\ 0 \\ 0 \\ 0 \end{bmatrix}. \tag{40}$$

This has no nonzero solutions. (Subtracting the second and third inequalities from the first gives $2\zeta x_1 \geqslant 0$, which combines with the fourth inequality to give $x_1 = 0$. With $x_1 = 0$, the first and third inequalities combine to give $x_2 = 0$.)

While further investigation of the joint design of the composition m and frame F—or of the product $D^{(m)} P F$ as P varies over the permutations induced by m—is merited, it is beyond the scope of this paper. Instead, we have extended our experiments with uniform sources to show the potential benefit of using entropy coding to exploit the lack of equiprobable codewords.

Fig. 4 summarizes experiments similar to those reported in Figs. 2 and 3. Each curve in this figure shows, for any given rate R on the horizontal axis, the lowest distortion can be achieved at any rate not exceeding R. The source $x \in \mathbb{R}^4$ has
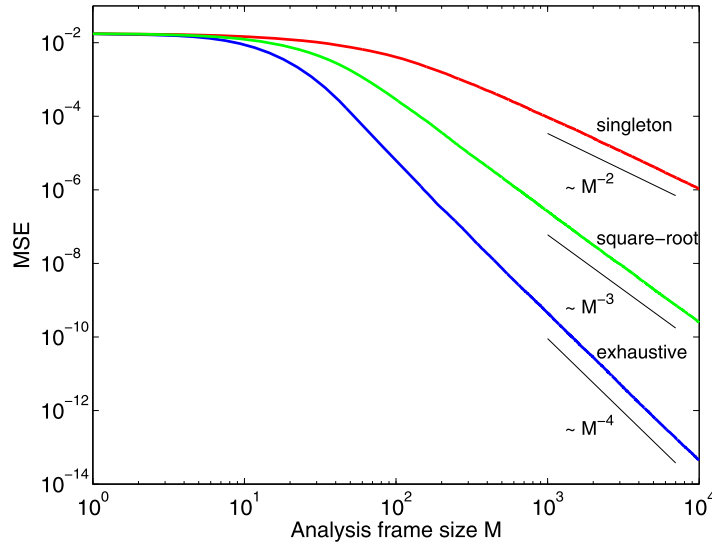
Fig. 5. Performance of recursive estimation for Variant I FPQ. The signal is of dimension $N = 8$ and various index sets $\mathcal{J}_k$ are used.

i.i.d. components uniformly distributed on $[-\frac{1}{2}, \frac{1}{2}]$, and Variant I FPQ with modulated harmonic tight frames of sizes $M = 6$ and 7 were used. Performance with rate measured only by (15) as before is labeled `fixed rate`. The codewords are highly nonequiprobable at all but the lowest rates. To demonstrate this, we alternatively measure rate by the empirical output entropy and label the performance `variable rate`. Clearly, the rate is significantly reduced by entropy coding at all but the lowest rates.

6.3. Recursive estimation experiments

The recursive estimation technique detailed in Algorithm 5 remains computationally feasible for large N and M. Here we simulate it with $x \in \mathbb{R}^N$ a nonrandom unit vector and $\{\phi_k\}_{k=1}^{\infty}$ an i.i.d. sequence of vectors drawn from the uniform distribution on the unit sphere in $\mathbb{R}^N$. Several choices for the $\mathcal{J}_k$ sets are used:

- *Singleton sets*: $\mathcal{J}_k = \{k - 1\}$;
- *Square-root sets*: $\mathcal{J}_k \subset \{1, 2, \ldots, k - 1\}$ is chosen uniformly at random from subsets of size $\lfloor \sqrt{k} \rfloor$; and
- *Exhaustive sets*: $\mathcal{J}_k = \{1, 2, \ldots, k - 1\}$.

Fig. 5 shows the sample mean estimate of $N^{-1}E[\|x - \hat{x}\|^2]$ over 1000 trials with $N = 8$ and M up to 10 000.

With singleton sets, we expect to see $\|x - \hat{x}_M\|^2 = \Theta(M^{-2})$ when M is increased without bound for any fixed N; Theorem 4.3 gives an upper bound of this order, and related lower bound results include [22, Thm. 6.1] and [8, Thm. 3]. With exhaustive sets, the total number of projections is $\frac{1}{2}M(M - 1)$, so squared error decay with the square number of projections would give $\|x - \hat{x}_M\|^2 = \Theta(M^{-4})$, and we indeed see this. Similarly, the empirical behavior is $\|x - \hat{x}_M\|^2 = \Theta(M^{-3})$ with square-root sets.

7. Proofs

7.1. Proof of Proposition 3.7

In order to prove Proposition 3.7, we need the following lemmas.

Lemma 7.1. *Assume that $M = N + 1$ and let $W = e^{j2\pi/M}$. Then for all $\alpha \in \mathbb{R}$ we have*

$$\sum_{i=1}^{N/2} W^{\alpha(2i-1)/2} = \frac{(-1)^\alpha - W^{\alpha/2}}{W^\alpha - 1}, \quad \textit{if } N \textit{ is even;}$$

and

$$\sum_{i=1}^{(N-1)/2} W^{\alpha i} = \frac{(-1)^\alpha - W^\alpha}{W^\alpha - 1}, \quad \textit{if } N \textit{ is odd.}$$

Proof. By noting that $W^{M/2} = -1$, we have the following computations:
N even:

$$\sum_{i=1}^{N/2} W^{\alpha(2i-1)/2} = W^{-\alpha/2} \cdot \sum_{i=1}^{N/2} W^{\alpha i} = W^{-\alpha/2} \cdot \frac{W^{\alpha(N+2)/2} - W^\alpha}{W^\alpha - 1}$$

$$= W^{-\alpha/2} \cdot \frac{(W^{M/2})^\alpha W^{\alpha/2} - W^\alpha}{W^\alpha - 1} = \frac{(-1)^\alpha - W^{\alpha/2}}{W^\alpha - 1}.$$

N odd:

$$\sum_{i=1}^{(N-1)/2} W^{\alpha i} = \frac{W^{\alpha(N+1)/2} - W^\alpha}{W^\alpha - 1} = \frac{(W^{M/2})^\alpha - W^\alpha}{W^\alpha - 1} = \frac{(-1)^\alpha - W^\alpha}{W^\alpha - 1}. \quad \square$$

Lemma 7.2. *For $M = N + 1$, the HTF $\Phi = \{\phi_k\}_{k=1}^M$ satisfies $\langle \phi_k, \phi_\ell \rangle = (-1)^{k-\ell+1}/N$, for all $1 \leqslant k < \ell \leqslant M$.*

Proof. Consider the two following cases:
N even: Using Euler's formula, for $k = 0, 1, \ldots, M-1$, ϕ_{k+1}^* can be rewritten as

$$\sqrt{\frac{2}{N}} \Bigg[\frac{W^k + W^{-k}}{2}, \frac{W^{3k} + W^{-3k}}{2}, \ldots, \frac{W^{(N-1)k} + W^{-(N-1)k}}{2},$$

$$\frac{W^k - W^{-k}}{2j}, \frac{W^{3k} - W^{-3k}}{2j}, \ldots, \frac{W^{(N-1)k} - W^{-(N-1)k}}{2j} \Bigg].$$

For $1 \leqslant k < \ell \leqslant M$, let $\alpha = k - \ell$. After some algebraic manipulations we can obtain

$$N \cdot \langle \phi_k, \phi_\ell \rangle = \sum_{i=1}^{N/2} W^{(k-\ell)(2i-1)/2} + \sum_{i=1}^{N/2} W^{(\ell-k)(2i-1)/2} = \frac{(-1)^\alpha - W^{\alpha/2}}{W^\alpha - 1} + \frac{(-1)^{-\alpha} - W^{-\alpha/2}}{W^{-\alpha} - 1}$$

$$= \frac{(-1)^\alpha W^{-\alpha/2} - 1}{W^{\alpha/2} - W^{-\alpha/2}} - \frac{(-1)^\alpha W^{\alpha/2} - 1}{W^{\alpha/2} - W^{-\alpha/2}} = (-1)^{\alpha+1}, \tag{41}$$

where (41) is obtained using Lemma 7.1.

N odd: Similarly, for $1 \leqslant k < \ell \leqslant M$ and $\alpha = k - \ell$, we have

$$N \cdot \langle \phi_k, \phi_\ell \rangle = 1 + \sum_{i=1}^{(N-1)/2} W^{(k-\ell)i} + \sum_{i=1}^{(N-1)/2} W^{(\ell-k)i} = 1 + \frac{(-1)^\alpha - W^\alpha}{W^\alpha - 1} + \frac{(-1)^{-\alpha} - W^{-\alpha}}{W^{-\alpha} - 1}$$

$$= 1 + \frac{(-1)^\alpha W^{-\alpha/2} - W^{\alpha/2}}{W^{\alpha/2} - W^{-\alpha/2}} - \frac{(-1)^\alpha W^{\alpha/2} - W^{-\alpha/2}}{W^{\alpha/2} - W^{-\alpha/2}} = 1 - (-1)^\alpha - 1 = (-1)^{\alpha+1}, \tag{42}$$

where (42) is due to Lemma 7.1. $\square$

Proof of Proposition 3.7. For a modulated HTF $\Psi = \{\psi_k\}_{k=1}^M$, as defined in Definition 3.4, for all $1 \leqslant k < \ell \leqslant M$ we have

$$\langle \psi_k, \psi_\ell \rangle = \langle \gamma(-1)^k \phi_k, \gamma(-1)^\ell \phi_\ell \rangle$$

$$= \gamma^2 (-1)^{k+\ell} (-1)^{k-\ell+1}/N \tag{43}$$

$$= (-1)^{k+\ell} (-1)^{k-\ell+1}/N \tag{44}$$

$$= -1/N, \tag{45}$$

where (43) is due to Lemma 7.2; and (44) is true because $|\gamma| = 1$ for all $1 \leqslant k < \ell \leqslant M$. Since the inner product is preserved through an orthogonal mapping, (45) is true for Type I and/or Type II equivalences of modulated HTFs as well. The tightness and unit norm of the HTF are obviously preserved for Type I and/or Type II equivalences. Therefore, the modulated HTFs and their equivalences of Type I and/or Type II are all restricted ETFs.

Conversely, from Proposition 3.5, every restricted ETF $\Psi = \{\psi\}_{k=1}^M$ can be represented up to Type I and Type II equivalences as follows:

$$\psi_k = \delta(k)\phi_k, \quad \text{for all } 1 \leqslant k \leqslant M,$$

where $\delta(k) = \pm 1$ is some sign function on k. Thus, the constraint $\langle \psi_k, \psi_\ell \rangle = a$ for some constant a of a restricted ETF is equivalent to

$$aN = N\delta(k)\delta(\ell) \cdot \langle \phi_k, \phi_\ell \rangle = \delta(k)\delta(\ell)(-1)^{k-\ell+1}, \quad \text{for all } 1 \leqslant k < \ell \leqslant M.$$

Therefore, $\delta(k)\delta(\ell)(-1)^{k-\ell}$ is constant for all $1 \leqslant k < \ell \leqslant M$. If we fix k and vary ℓ, it is clear that the sign of $\delta(\ell)$ must be alternatingly changed. Thus, Ψ is one of the two HTFs specified in the proposition, completing the proof. $\quad\square$

7.2. Proof of Theorem 5.3

The following lemmas are all stated for Variant II initial codewords. They are somewhat stronger than what we need for the proof of Theorem 5.3 because a Variant II initial codeword is automatically a Variant I initial codeword. However, these lemmas will be reused to prove Theorem 5.5 in Section 7.3.

For convenience, if $\{i_1, \ldots i_k\}$ is a subset of $\{1, 2, \ldots, M\}$ and σ is a permutation on that subset, we simply write

$$P = \begin{pmatrix} i_1 & i_2 & \cdots i_k \\ \sigma(i_1) & \sigma(i_2) & \cdots \sigma(i_k) \end{pmatrix}$$

if Py maps y_{i_ℓ} to $y_{\sigma(i_\ell)}$, $1 \leqslant \ell \leqslant k$, and fixes all the other components of vector y. This notation with parentheses should not be confused with a matrix, for which we always use square brackets.

Proofs of the lemmas rely heavily on the key observation that the operator $P(\cdot)P^{-1}$ first permutes the columns of the original matrix, then permutes the rows of the resulting matrix in the same manner.

Lemma 7.3. *Assume that $M \geqslant 3$. If the entries of matrix A satisfy $a_{k,1} \neq a_{\ell,1}$ for some $1 < k < \ell$, then there exists a pair $(P, \hat{y}_{\text{init}})$, where P is a permutation matrix and $\hat{y}_{\text{init}}$ is a Variant II initial codeword compatible with some composition, such that the inequality (31) is violated.*

Proof. Consider the two following cases:

Case 1: If $a_{k,1} < a_{\ell,1}$, choose $P = I_M$, and $\hat{y}_{\text{init}} = (\mu_1, \mu_2, \ldots, \mu_M)$. Consider the following difference:

$$\Delta_{k,\ell} = \langle (a_{k,j})_j, \hat{y}_{\text{init}} \rangle - \langle (a_{\ell,j})_j, \hat{y}_{\text{init}} \rangle = \sum_{j=1}^{M} a_{k,j}\mu_j - \sum_{j=1}^{M} a_{\ell,j}\mu_j$$

$$= (a_{k,1} - a_{\ell,1})\mu_1 + \left(\sum_{j=2}^{M} a_{k,j}\mu_j - \sum_{j=2}^{M} a_{\ell,j}\mu_j \right).$$

Fix $\mu_2 > \mu_3 > \cdots > \mu_M \geqslant 0$ and let μ_1 go to $+\infty$. Since $a_{k,1} < a_{\ell,1}$, $\Delta_{k,\ell}$ will go to $-\infty$. Thus, there exist $\mu_1 > \mu_2 > \cdots > \mu_M \geqslant 0$ such that $\Delta_{k,\ell} < 0$. On the other hand, for $m = (1, 1, \ldots, 1)$, inequality (31) requires that $\Delta_{k,\ell} \geqslant 0$ for all $k < \ell$. Therefore the chosen pair violates inequality (31).

Case 2: If $a_{k,1} > a_{\ell,1}$, choose $P = \begin{pmatrix} k & \ell \\ \ell & k \end{pmatrix}$. Since $k, \ell \neq 1$, the entries of matrix $A' = PAP^{-1}$ will satisfy $a'_{k,1} < a'_{\ell,1}$. We return to the first case, completing the proof. $\quad\square$

Lemma 7.4. *Assume that $M \geqslant 3$. If the entries of matrix A satisfy $a_{k,j} \neq a_{\ell,j}$, for any pairwise distinct triple (k, j, ℓ), then there exists a pair $(P, \hat{y}_{\text{init}})$, where P is a permutation matrix and $\hat{y}_{\text{init}}$ is a Variant II initial codeword compatible with some composition, such that the inequality (31) is violated.*

Proof. We first show that there exists some permutation matrix P_1 such that $\tilde{A} = P_1 A P_1^{-1}$ satisfies the hypothesis of Lemma 7.3. Indeed, consider the following cases:

1. If $j = 1$, it is obvious to choose $P_1 = I_M$.
2. If $j > 1$ and $k > 1$, choosing $P_1 = \begin{pmatrix} 1 & j \\ j & 1 \end{pmatrix}$ yields $\tilde{a}_{k,1} = a_{k,j} \neq a_{\ell,j} = \tilde{a}_{\ell,1}$, since $k, \ell \notin \{1, j\}$.
3. If $j > 1$ and $k = 1$, choosing $P_1 = \begin{pmatrix} 1 & j \\ j & 1 \end{pmatrix}$ yields $\tilde{a}_{j,1} = a_{k,j} \neq a_{\ell,j} = \tilde{a}_{\ell,1}$, since $k = 1$, and $\ell \notin \{1, j\}$. Note that in this case, $j \neq 1$, and so $\tilde{A}$ satisfies the hypothesis of Lemma 7.3.

Now with P_1 chosen as above, according to Lemma 7.3 there exists a pair $(P_2, \hat{y}_{\text{init}})$, where P is a permutation matrix and $\hat{y}_{\text{init}}$ is a Variant II initial codeword compatible with some composition, such that

$$\mathbf{0} \not\leq D^{(m)} P_2 \tilde{A} P_2^{-1} \hat{y}_{\text{init}} = D^{(m)} P_2 \left(P_1 A P_1^{-1} \right) P_2^{-1} \hat{y}_{\text{init}} = D^{(m)} PAP^{-1} \hat{y}_{\text{init}},$$

where $P \triangleq P_2 P_1$. Since the product of any two permutation matrices is also a permutation matrix, the pair $(P, \hat{y}_{\text{init}})$ violates the inequality (31). $\quad\square$

Lemma 7.5. *Suppose that A is a diagonal matrix. Then the inequality* (31) *holds for every composition and every Variant* II *initial codeword compatible with it, only if A is equal to the identity matrix scaled by a nonnegative factor.*

Proof. Suppose that $A = \text{diag}(a_1, a_2, \ldots, a_M)$. We first show that $a_i \geqslant 0$ for every i by contradiction.

If $a_1 < 0$, we can choose $P = I_M$ and $\mu_1 > \mu_2 > \cdots > \mu_M \geqslant 0$, where μ_1 is large enough relative to $\mu_2, \ldots, \mu_M$ to violate inequality (31).

If $a_j < 0$ for some $1 < j \leqslant M$, using $P = \left(\begin{smallmatrix} 1 & j \\ j & 1 \end{smallmatrix}\right)$ yields $a'_1 = a_j < 0$, where a'_1 is the first entry on the diagonal of matrix $A' \triangleq PAP^{-1}$. Repeating the previous argument, we get the contradiction.

Now we show that if $a_k \neq a_\ell$ for some $1 \leqslant k < \ell \leqslant M$, there exists a pair $(P, \hat{y}_{\text{init}})$, where P is a permutation matrix and $\hat{y}_{\text{init}}$ is a Variant II initial codeword compatible with some composition, such that inequality (31) is violated.

Case 1: if $a_k < a_\ell$, choose $P = I_M$ and consider $\hat{y}_{\text{init}} = (\mu_1, \mu_2, \ldots, \mu_M)$, where $\mu_\ell = \mu_k - \varepsilon$ for some positive ε. Choose μ_k such that

$$\mu_k > \frac{\varepsilon a_\ell}{a_\ell - a_k} \geqslant 0. \tag{46}$$

On the other hand, we can choose ε small enough so that μ_ℓ is positive as well. The other components can therefore be chosen to make $\hat{y}_{\text{init}}$ a Variant II initial codeword compatible with composition $m = (1, 1, \ldots, 1)$. For the above choice of μ_k we can easily check that $\Delta_{k,\ell} = a_k \mu_k - a_\ell \mu_\ell < 0$, violating inequality (31).

Case 2: if $a_k > a_\ell$, choosing $P = \left(\begin{smallmatrix} k & \ell \\ \ell & k \end{smallmatrix}\right)$ yields $PAP^{-1} = \text{diag}(a_1, a_2, \ldots, a_\ell, \ldots, a_k, \ldots, a_M)$.

We return to case 1, completing the proof. □

Proof of Theorem 5.3. First note that a Variant II initial codeword is always a Variant I initial codeword, therefore, Lemmas 7.3, 7.4, and 7.5 also apply for Variant I initial codewords. From Lemma 7.4, all entries on each column of matrix A are constant except for the one that lies on the diagonal. Thus, A can be written as $A = \tilde{I} + J$, where $\tilde{I} = \text{diag}(a_1, a_2, \ldots, a_M)$, and

$$J = \begin{bmatrix} b_1 & b_2 & \cdots & b_M \\ b_1 & b_2 & \cdots & b_M \\ \vdots & \vdots & & \vdots \\ b_1 & b_2 & \cdots & b_M \end{bmatrix} \in \mathcal{J}.$$

Recall that from properties (P1) and (P2) of J we have

$$D^{(m)} P J P^{-1} = \mathbf{0}, \quad \text{for any } m.$$

Hence,

$$D^{(m)} P \tilde{I} P^{-1} \hat{y}_{\text{init}} \geqslant \mathbf{0}, \quad \text{for any } m \text{ and any } \hat{y}_{\text{init}}. \tag{47}$$

From (47) and Lemma 7.5, we can deduce that $\tilde{I} = a I_M$, for some nonnegative constant a. □

7.3. Proof of Theorem 5.5

In order for R to produce consistent reconstructions, we need the following inequality (noting that $V = V^{-1}$ for any $V \in \mathcal{Q}(m)$):

$$\widetilde{D}^{(m)} V P A P^{-1} V \hat{y}_{\text{init}} \geqslant \mathbf{0}, \quad \text{for any } V \in \mathcal{Q}(m) \text{ and } P \in \mathcal{G}(m), \tag{48}$$

where $A = FR$. We first fix the sign-changing matrix V to be the identity matrix I_M. Then the first $L(m)$ rows of (48) exactly form the inequality (31). Since Lemmas 7.3, 7.4, and 7.5 are stated for Variant II initial codewords, it follows from Theorem 5.3 that A must be of the form $a I_M + J$, where $a \geqslant 0$ and $J \in \mathcal{J}$. Substituting in to (48), we obtain

$$a\widetilde{D}^{(m)} \hat{y}_{\text{init}} + \widetilde{D}^{(m)} V P J P^{-1} V \hat{y}_{\text{init}} \geqslant \mathbf{0}. \tag{49}$$

Now we show that $J = \mathbf{0}$ by contradiction. Suppose all entries in column i of J are b_i, for $1 \leqslant i \leqslant M$. Consider the following cases:

1. If b_1 is negative, choose $V = P = I_M$ and $\hat{y}_{\text{init}} = (\mu_1, \mu_2, \ldots, \mu_M)$ compatible with composition $m = (1, 1, \ldots, 1)$. Consider the last row of inequality (49):

$$b_1 \mu_1 + a \mu_{M-1} + \sum_{i=2}^{M} b_i \mu_i \geqslant 0. \tag{50}$$

Since $M \geqslant 3$, $M - 1 \neq 1$. Therefore the scale associated with μ_1 in the left-hand side of inequality (50) is $b_1 < 0$. Hence, choosing μ_1 large enough certainly breaks inequality (50), and therefore violates inequality (49).

2. If b_1 is positive, choosing $P = I_M$, $V = \text{diag}(-1, 1, 1, \ldots, 1)$ makes the first entry of the $(M - 1)$th row of matrix $VPJP^{-1}V$ negative (note that $M - 1 \neq 1$ and the operator $V(\cdot)V$ first changes the signs of columns of the original matrix and then changes the signs of rows of the resulting matrix in the same manner). Repeating the argument in the first case we can break the last row of inequality (49) by appropriate choice of $\hat{y}_{\text{init}}$.

3. If column ℓ of J, $1 < \ell \leqslant M$, is different from zero, choosing $P = \left(\begin{smallmatrix} 1 & \ell \\ \ell & 1 \end{smallmatrix} \right)$ leads us to either case 1 or case 2.

Hence,

$$A = FR = aI_M. \tag{51}$$

Equality (51) states that the row vectors of F and the column vectors of R form a biorthogonal basis pair of $\mathbb{R}^N$ within a nonnegative scale factor. Since the number of vectors in each basis cannot exceed the dimension of the space, we can deduce $M \leqslant N$. On the other hand, $M \geqslant N$ because F is a frame. Thus, $M = N$.

Acknowledgments

References

[1] J. Kovačević, A. Chebira, Life after bases: The advent of frames (Part I), IEEE Signal Process. Mag. 24 (4) (2007) 86–104.
[2] J. Kovačević, A. Chebira, Life after bases: The advent of frames (Part II), IEEE Signal Process. Mag. 24 (5) (2007) 115–125.
[3] I. Daubechies, Ten Lectures on Wavelets, SIAM, Philadelphia, PA, 1992.
[4] N.T. Thao, M. Vetterli, Reduction of the MSE in R-times oversampled A/D conversion from $O(1/R)$ to $O(1/R^2)$, IEEE Trans. Signal Process. 42 (1994) 200–203.
[5] N.T. Thao, M. Vetterli, Deterministic analysis of oversampled A/D conversion and decoding improvement based on consistent estimates, IEEE Trans. Signal Process. 42 (3) (1994) 519–531.
[6] V.K. Goyal, M. Vetterli, N.T. Thao, Quantized overcomplete expansions in $\mathbb{R}^N$: Analysis, synthesis, and algorithms, IEEE Trans. Inform. Theory 44 (1) (1998) 16–31.
[7] Z. Cvetković, Resilience properties of redundant expansions under additive noise and quantization, IEEE Trans. Inform. Theory 49 (3) (2003) 644–656.
[8] S. Rangan, V.K. Goyal, Recursive consistent estimation with bounded noise, IEEE Trans. Inform. Theory 47 (1) (2001) 457–464.
[9] J.J. Benedetto, A.M. Powell, Ö. Yilmaz, Sigma–Delta ($\Sigma\Delta$) quantization and finite frames, IEEE Trans. Inform. Theory 52 (5) (2006) 1990–2005.
[10] B.G. Bodmann, V.I. Paulsen, Frame paths and error bounds for sigma–delta quantization, Appl. Comput. Harmon. Anal. 22 (2) (2007) 176–197.
[11] B.G. Bodmann, S.P. Lipshitz, Randomly dithered quantization and sigma–delta noise shaping for finite frames, Appl. Comput. Harmon. Anal. 25 (3) (2008) 367–380.
[12] A.M. Powell, Mean squared error bounds for the Rangan–Goyal soft thresholding algorithm, Appl. Comput. Harmon. Anal. 29 (3) (2010) 251–271.
[13] B. Beferull-Lozano, A. Ortega, Efficient quantization for overcomplete expansions in $\mathbb{R}^n$, IEEE Trans. Inform. Theory 49 (1) (2003) 129–150.
[14] J.G. Dunn, Coding for continuous sources and channels, PhD thesis, Columbia University, New York, 1965.
[15] T. Berger, F. Jelinek, J.K. Wolf, Permutation codes for sources, IEEE Trans. Inform. Theory IT-18 (1972) 160–169.
[16] V. Brajovic, T. Kanade, A VLSI sorting image sensor: Global massively parallel intensity-to-time processing for low-latency adaptive vision, IEEE Trans. Robot. Automat. 15 (1) (1999) 67–75.
[17] X. Guo, X. Qi, J.G. Harris, A time-to-first-spike CMOS image sensor, IEEE Sensors J. 7 (8) (2007) 1165–1175.
[18] Z. Cvetković, M. Vetterli, Error-rate characteristics of oversampled analog-to-digital conversion, IEEE Trans. Inform. Theory 44 (5) (1998) 1961–1964.
[19] Z. Cvetković, I. Daubechies, B.F. Logan, Single-bit oversampled A/D conversion with exponential accuracy in the bit rate, IEEE Trans. Inform. Theory 53 (11) (2007) 3979–3989.
[20] D.L. Donoho, M. Vetterli, R.A. DeVore, I. Daubechies, Data compression and harmonic analysis, IEEE Trans. Inform. Theory 44 (6) (1998) 2435–2476.
[21] L.R. Varshney, V.K. Goyal, Ordered and disordered source coding, in: Proc. UCSD Workshop Inform. Theory Its Applications, La Jolla, CA, 2006.
[22] N.T. Thao, M. Vetterli, Lower bound on the mean-squared error in oversampled quantization of periodic signals using vector quantization analysis, IEEE Trans. Inform. Theory 42 (2) (1996) 469–479.
[23] R.M. Gray, T.G. Stockham Jr., Dithered quantizers, IEEE Trans. Inform. Theory 39 (3) (1993) 805–812.
[24] A. Gersho, R.M. Gray, Vector Quantization and Signal Compression, Kluwer Academic Publishers, 1992.
[25] D. Slepian, Permutation modulation, Proc. IEEE 53 (3) (1965) 228–236.
[26] T. Berger, Optimum quantizers and permutation codes, IEEE Trans. Inform. Theory IT-18 (1972) 759–765.
[27] T. Berger, Minimum entropy quantizers and permutation codes, IEEE Trans. Inform. Theory IT-18 (1982) 149–157.
[28] D. Slepian, Group codes for the Gaussian channel, Bell System Tech. J. 47 (1968) 575–602.
[29] F. Jelinek, Buffer overflow in variable length coding of fixed rate sources, IEEE Trans. Inform. Theory IT-14 (1968) 490–501.
[30] T.M. Cover, Enumerative source coding, IEEE Trans. Inform. Theory IT-19 (1) (1973) 73–77.
[31] H.A. David, H.N. Nagaraja, Order Statistics, 3rd edition, Wiley–Interscience, 2003.
[32] V.K. Goyal, S.A. Savari, W. Wang, On optimal permutation codes, IEEE Trans. Inform. Theory 47 (2001) 2961–2971.
[33] D. Han, D.R. Larson, Frames, Bases and Group Representations, American Mathematical Society, Providence, RI, 2000.
[34] V.K. Goyal, J. Kovačević, J.A. Kelner, Quantized frame expansions with erasures, Appl. Comput. Harmon. Anal. 10 (3) (2001) 203–233.
[35] Y. Eldar, H. Bölcskei, Geometrically uniform frames, IEEE Trans. Inform. Theory 49 (4) (2003) 993–1006.
[36] J.D. Kolesar, SD modulation and correlation criteria for the construction of finite frames arising in communication theory, PhD thesis, University of Maryland, College Park, MD, 2004.
[37] F. Abdelkefi, Performance of Sigma–Delta quantizations in finite frames, IEEE Trans. Inform. Theory 54 (11) (2008) 5087–5101.
[38] R.B. Holmes, V.I. Paulsen, Optimal frames for erasures, Linear Algebra Appl. 377 (2004) 31–51.
[39] T. Strohmer, R.W. Heath, Grassmannian frames with application to coding and communications, Appl. Comput. Harmon. Anal. 14 (3) (2003) 257–275.

[40] M.A. Sustik, J.A. Tropp, I.S. Dhillon, R.W. Heath, On the existence of equiangular tight frames, Linear Algebra Appl. 426 (2007) 619–635.
[41] J.A. Tropp, Greed is good: Algorithmic results for sparse approximation, IEEE Trans. Inform. Theory 50 (10) (2004) 2231–2242.
[42] T. Strohmer, A note on equiangular tight frames, Linear Algebra Appl. 429 (2008) 326–330.
[43] P.G. Casazza, D. Redmond, J.C. Tremain, Real equiangular frames, in: Proc. 42nd Annu. Conf. Information Sciences and Systems (CISS 2008), 2008, pp. 715–720.
[44] H. Viswanathan, R. Zamir, On the whiteness of high-resolution quantization errors, IEEE Trans. Inform. Theory 47 (5) (2001) 2029–2038.
[45] D.J. Sakrison, A geometric treatment of the source encoding of a Gaussian random variable, IEEE Trans. Inform. Theory IT-14 (1968) 481–486.
[46] A. György, T. Linder, Optimal entropy-constrained scalar quantization of a uniform source, IEEE Trans. Inform. Theory 46 (7) (2000) 2704–2711.
[47] H.Q. Nguyen, L.R. Varshney, V.K. Goyal, Concentric permutation source codes, IEEE Trans. Commun. 58 (11) (2010).
[48] D. Marco, D.L. Neuhoff, Low-resolution scalar quantization for Gaussian sources and squared error, IEEE Trans. Inform. Theory 52 (4) (2006) 1689–1697.