Simultaneous Segmentation, Compression, and Denoising of Signals using Polyharmonic Local Sine Transform and Minimum Description Length Criterion

Naoki Saito, *Senior Member, IEEE* and Ernest Woei

Abstract

We present a new approach to simultaneously segment, compress, and denoise an observed noisy signal by combining our compact signal representation scheme called the *Polyharmonic Local Sine Transform* (PHLST) and the Minimum Description Length (MDL) criterion. The PHLST algorithm first generates a redundant set of local pieces of an input signal each of which is supported on a dyadic subinterval and is approximated by a combination of an algebraic polynomial of low order and a trigonometric polynomial. This combination of polynomials compensates their shortcomings and yields a compact representation of the local piece. To select the best nonredundant combination of the local pieces from this redundant set, we use the MDL criterion with or without actually quantizing the relevant parameters. The resulting representation gives rise to simultaneous segmentation, compression, and denoising of the given data. We apply our algorithms to synthetic and real datasets and compare their performance against other competing methods for denoising and compression such as the wavelet transform using the MDL criterion. We observe that our PHLST algorithms perform better (in compression rate, relative ℓ^2-error, and visual quality) than the wavelet transform for oscillatory signals whereas their performance is comparable to that of the wavelet transform for piecewise smooth signals.

Index Terms

Polyharmonic Local Sine Transform, Signal Compression, Denoising, Quantization, MDL, Piecewise Approximation

The authors are with Department of Mathematics, University of California, Davis, One Shields Avenue, Davis, CA 95616.

I. INTRODUCTION

For signal compression and feature extraction purposes, it is of significant interest to segment the input data according to the local smoothness and the geometry of the singularities. There is no need to subdivide a smooth region into a set of many smaller segments, and in fact, that is wasteful because each segment requires to store some information such as the endpoints of the segment. This was also demonstrated by our earlier papers [1], [2] using the so-called *polyharmonic local sine transform* (PHLST). The original form of PHLST, however, assumes that the partition of an input signal is given a priori, and does not automatically compute the best possible partition for the signal. In this paper, we propose an automatic method to do that for 1D signals and give several convincing examples. Our approach is based on the "split-and-merge" or "divide-and-conquer" strategy à la best basis of Coifman and Wickerhauser [3]. We first split (or subdivide) an input signal brutally into a set of local pieces by multiplying the characteristic functions supported on dyadic subintervals in the form of a binary tree. Then, we represent each local piece using the PHLST and evaluate its cost in terms of *Minimum Description Length* (*MDL*) criterion. Finally, we "prune" this tree to come up with the "best" split or segmentation of the original signal, which results in the minimum overall MDL cost.

There are several published works closely related to our project. In [4] we showed how we used a library of orthonormal bases and the MDL criterion to give the best compromise between the fidelity of the estimation result to the data and the efficiency of the representation of the estimated signal: it selects the "best" basis and the "best" number of terms to be retained out of the various bases provided in the library in an objective manner. The significant difference between this paper and [4] is that we now select the best representation from scratch while in [4] we simply used the MDL criterion to choose the best number of terms to be retained in the pre-computed best basis that was selected by a different criterion, e.g., the minimum entropy criterion [3].

Moulin [5] applied the idea of adapted tree structures in a wavelet packet library by viewing the choice of a tree as a choice between competing models, and choosing the best model according to the MDL principle. In fact, we adopt his idea of node selection cost in our algorithm as we will discuss in Section VI. However, there are at least two differences between his approach and our approach. First, our approach is not for wavelet packets. It is designed for our PHLST representation of a signal. Second, one of our two proposed methods quantizes all the relevant parameters to convert them into integers followed by the MDL cost computation whereas Moulin's approach does not use the quantization procedure.

To improve denoising performance, Hansen and Yu [6] folded the prior distributional assumptions for

natural images into a model selection framework for wavelet denoising via MDL. Another important aspect of their work is their clear understanding on the difference between the signal models with and without quantizing the parameters (e.g., wavelet coefficients) used in the models. We also distinguish these two models and propose the corresponding algorithms using the PHLST representation of a given signal. Hansen and Yu, however, strictly used a fixed wavelet basis selected by a user and their algorithm is not designed to choose an optimal basis from a library of orthonormal bases.

The organization of this paper is the following. We will discuss two different versions of our basic formulation for simultaneous compression and denoising in Sections IV and V. Section VI further develops our algorithm for signal segmentation, which will be followed by our numerical experiments in Section VII. We will then conclude this paper in Section VIII. But first, let us review briefly our PHLST scheme in Section II and set up our noisy signal model in Section III.

II. Review of PHLST

We will review the one-dimensional and global version (i.e., without subdividing the domain) of our PHLST scheme. For higher dimensions and the details, see [1], [2]. Suppose our signal $f(x)$ is supported on the unit interval $I = [0, 1]$, and has some smoothness, e.g., $f \in C^{2m}(I)$ for some $m \in \mathbb{N}$. Now, we separate the data function into two pieces

$$f(x) = u(x) + v(x). \tag{1}$$

The *polyharmonic* component u in (1) satisfies the following *polyharmonic differential equation*.

$$\frac{\mathrm{d}^{2m}u}{\mathrm{d}\,x^{2m}}(x) = u^{(2m)}(x) = 0, \quad x \in I, \tag{2}$$

with the boundary condition

$$u^{(2\ell)}(0) = f^{(2\ell)}(0), u^{(2\ell)}(1) = f^{(2\ell)}(1), \quad 0 \le \ell < m. \tag{3}$$

The u component satisfying the above conditions is a $2m-1$ degree algebraic polynomial with $2m$ coefficients. We note that in higher dimensions, the polyharmonic equation (2) becomes $\Delta^m u = 0$ where Δ is the Laplace operator and its solution is not an algebraic polynomial in general.

Once we compute the u component, then the *residual* $v = f - u$ is computed and expanded into the Fourier *sine* series

$$v(x) = \sqrt{2}\sum_{\ell=1}^{\infty} \beta_\ell \sin(\pi \ell x), \quad \beta_\ell = \sqrt{2}\int_0^1 v(x)\sin(\pi\ell x)\,\mathrm{d}x.$$

Thanks to the boundary condition (3), the v component satisfies

$$v^{(2\ell)}(0) = v^{(2\ell)}(1) = 0, \quad 0 \le \ell < m,$$

which makes the Fourier sine coefficients decay very quickly, i.e., $|\beta_\ell| \approx O(\ell^{-2m-1})$. One can compare this decay rate with that of the ordinary Fourier series expansion with the periodic boundary condition, which gives rise to $O(\ell^{-1})$ with the infamous Gibbs phenomenon [7, Sec. 10], or that of the Fourier cosine series expansion with the Neumann boundary condition, which gives rise to $O(\ell^{-2})$. See [2] for the proof of the above fact and the details of the decay rates of these coefficients. The main point of the use of PHLST for signal compression is this speed of decay of the expansion coefficients. This means that we can truncate the coefficients with a smaller number of terms and still get a good approximation if the original signal has enough smoothness. Moreover, representing the u component only requires $2m$ real-valued numbers since it is an algebraic polynomial of degree $2m-1$. Another advantage of the PHLST representation is its usefulness for signal interpolation and derivative computation at arbitrary points in I thanks to the use of both the algebraic polynomial in u and the trigonometric polynomial in v. This combination also compensates each other's shortcomings. If we were to use only a trigonometric polynomial to approximate the data, we would encounter the Gibbs phenomenon. On the other hand, if we were to use only an algebraic polynomial (of high degree) to approximate the data, then we would encounter the so-called Runge phenomenon [7, Sec. 18] that results in totally erroneous interpolation.

Of course, the story gets more complicated (and interesting) in more realistic situations because those signals of our interest contain noise, singularities, and transients, which will be discussed below.

For the practical purposes, we only consider $m = 1$ in this paper. Then we have

$$u(x) = \alpha_0 + \alpha_1 x, \tag{4}$$

where the coefficients α_k's are determined from the boundary conditions.

Remark II.1. Note that for $m = 2$, we have a cubic polynomial: $u(x) = \alpha_0 + \alpha_1 x + \alpha_2 x^2 + \alpha_3 x^3$. To determine α_k's, we need to estimate $f''(0)$ and $f''(1)$ from the data. Although there are several interesting algorithms to estimate them (e.g., [7, Sec. 19]), which we are currently investigating, we will not deal with this cubic case in this paper. We believe that the cases with $m > 2$ are impractical due to the need of estimating even higher derivatives from the data.

III. Our Signal Model

Let us now consider the case where the data contain additive white Gaussian noise (AWGN) with unknown variance σ^2. Suppose the data are sampled uniformly at $x_n = n/N$, $n = 0, 1, \ldots, N$. Thus,

our signal model can be written as

$$f(x_n) = u(x_n) + v(x_n) + \eta(x_n), \quad \eta(x_n) \overset{\text{i.i.d.}}{\sim} \mathcal{N}(0, \sigma^2). \tag{5}$$

In the vector notation, (5) can be written as

$$\boldsymbol{f} = \boldsymbol{u} + \boldsymbol{v} + \boldsymbol{\eta} \in \mathbb{R}^{N+1}, \quad \boldsymbol{\eta} \sim \mathcal{N}(\boldsymbol{0}, \sigma^2 I_{N+1}). \tag{6}$$

We denote the kth entry of $\boldsymbol{f}$ by $\boldsymbol{f}[k]$. Thus $\boldsymbol{f}[0] = f(0)$ and $\boldsymbol{f}[N] = f(1)$. The $\boldsymbol{u}$ component can be written as

$$\boldsymbol{u} = R\boldsymbol{\alpha}, \quad R \triangleq \begin{bmatrix} 1 & 0 \\ 1 & \Delta x \\ \vdots & \vdots \\ 1 & N\Delta x \end{bmatrix} \in \mathbb{R}^{(N+1)\times 2}, \quad \Delta x = 1/N, \tag{7}$$

where $\boldsymbol{\alpha} = (\alpha_0, \alpha_1)^T$.

We assume that the $\boldsymbol{v}$ component consists of M sinusoids ($0 \le M \le N-1$) with frequencies $1 \le \nu_1, \ldots, \nu_M \le N-1$ instead of $N-1$ sinusoids of frequencies $1, \ldots, N-1$:

$$v(x_n) = \sqrt{\frac{2}{N}} \sum_{\ell=1}^{M} \beta_{\nu_\ell} \sin(\pi \nu_\ell x_n), \tag{8}$$

where

$$\beta_{\nu_\ell} = \sqrt{\frac{2}{N}} \sum_{n=1}^{N-1} v(x_n) \sin(\pi \nu_\ell x_n). \tag{9}$$

These are a subset of the Discrete Sine Transform (in fact the so-called DST Type I or DST-I for short) coefficients of the $\boldsymbol{v}$ component. The reason why we model the $\boldsymbol{v}$ component by M sinusoids instead of $N-1$ sinusoids is the following. The column vectors of U in (7) and the $N-1$ DST basis vectors jointly span the whole space $\mathbb{R}^{N+1}$, i.e., they can completely represent the given data including the noise without error. Hence, we could not reduce noise if we were to use all $N-1$ sinusoids. Let us write the $\boldsymbol{v}$ component as

$$\boldsymbol{v} = S\boldsymbol{\beta}, \quad S \triangleq \begin{bmatrix} 0 & \cdots & 0 \\ \vdots & S_{N-1} & \vdots \\ 0 & \cdots & 0 \end{bmatrix} \in \mathbb{R}^{(N+1)\times(N+1)}, \tag{10}$$

where S_{N-1} is the DST-I basis matrix of size $(N-1)\times(N-1)$. The coefficient vector $\boldsymbol{\beta}$ is of length $N+1$ but has at most M nonzero entries. Furthermore, $\boldsymbol{v}[0] = \boldsymbol{v}[N] = 0$ and $\boldsymbol{\beta}[0] = \boldsymbol{\beta}[N] = 0$ since the $\boldsymbol{u}$ component removes the endpoints. Therefore, our signal model (6) can be rewritten as

$$\boldsymbol{f} = R\boldsymbol{\alpha} + S\boldsymbol{\beta} + \boldsymbol{\eta}, \quad \boldsymbol{\eta} \sim \mathcal{N}(\boldsymbol{0}, \sigma^2 I_{N+1}), \tag{11}$$

which can be further simplified as

$$\boldsymbol{f} = W\boldsymbol{\gamma} + \boldsymbol{\eta}, \quad \boldsymbol{\eta} \sim \mathcal{N}(\mathbf{0}, \sigma^2 I_{N+1}) \tag{12}$$

by defining

$$W \triangleq [R(:,1) \,|\, S(:,2:N) \,|\, R(:,2)] \in \mathbb{R}^{(N+1)\times(N+1)}, \tag{13}$$

$$\boldsymbol{\gamma} \triangleq \begin{bmatrix} \alpha_0 \\ \boldsymbol{\beta}[1:N-1] \\ \alpha_1 \end{bmatrix} \in \mathbb{R}^{N+1}. \tag{14}$$

Note that the matrix W is *not orthogonal*.

IV. Analytical Compression and Denoising

The essence of MDL is the following. Suppose that we are given data $\boldsymbol{d} \in \mathbb{R}^n$ that were generated by some parametric statistical model $P(\boldsymbol{d}\,|\,\boldsymbol{\theta})$ where $\boldsymbol{\theta} \in \mathbb{R}^k$. Suppose also that we want to have a flexibility that k is not fixed a priori and want to learn the "best" k and $\boldsymbol{\theta}$ from the data. Rissanen advocates (see e.g., [8]) that the best model is the minimizer of the following cost functional (or codelength):

$$L(\boldsymbol{d}, \boldsymbol{\theta}) = L(\boldsymbol{d}\,|\,\boldsymbol{\theta}) + L(\boldsymbol{\theta}) \approx -\log P(\boldsymbol{d}\,|\,\boldsymbol{\theta}) + \frac{k}{2}\log n. \tag{15}$$

In this paper, $\log$ denotes the base 2 logarithm unless stated otherwise. The first term quantifies how well this model can fit the data. The second term is to penalize complicated models: the simpler the model (i.e., the smaller k), the better. MDL balances these two conflicting terms using the information theoretic justification.

There are two possible ways to apply to MDL criterion to our problem. One is called the "analytical" formulation, the other is called the "quantized" formulation (see also [6] for more about such terminology). In this section, we focus on the analytical formulation, which essentially uses the MDL criterion as a way to select the number of model parameters and to compute their maximum likelihood estimates (MLEs). We also used this strategy in our earlier paper [4] for signal compression and denoising using the wavelet packets and local trigonometric dictionaries. On the other hand, the quantized formulation (which we will discuss in Section V) actually performs the quantization of all the parameters, i.e., it truly converts everything into "bits" and seeks the model that generates the shortest bitstream for the given data. Let us now introduce our notation for the MDL formulation.

Let $\boldsymbol{\theta}_u$ and $\boldsymbol{\theta}_v$ be the vectors of parameters that completely specify the $\boldsymbol{u}$ and $\boldsymbol{v}$ components in (7) and (10), respectively. It is clear that $\boldsymbol{\theta}_u$ is simply the pair $(m, \boldsymbol{\alpha})$ if we have a choice in m, say $m = 1$

or 2. Since we only consider the $m = 1$ case, we can assume that $\boldsymbol{\theta}_u = \boldsymbol{\alpha} \in \mathbb{R}^2$. As for $\boldsymbol{\theta}_v$, it is a concatenation of M nonzero real-valued DST coefficients $(\beta_{\nu_1}, \ldots, \beta_{\nu_M})^T \in \mathbb{R}^M$ and their indicator vector $(\nu_1, \ldots, \nu_M)^T \in \{1, \ldots, N-1\}^M$. Therefore, the codelength (15) of our data with our signal model (6) (or equivalently (11) or (12)) can be written as

$$L(\boldsymbol{f}, \boldsymbol{\theta}_u, \boldsymbol{\theta}_v, \sigma^2) = L(\sigma^2) + L(\boldsymbol{\theta}_u, \boldsymbol{\theta}_v \,|\, \sigma^2) + L(\boldsymbol{f} \,|\, \boldsymbol{\theta}_u, \boldsymbol{\theta}_v, \sigma^2) \tag{16}$$

Note that we *need* to estimate all these parameters via the maximum likelihood method. Let $\widehat{\theta}$ be the MLE of a parameter θ. By the definition of MLE, we have

$$L(\boldsymbol{f}, \boldsymbol{\theta}_u, \boldsymbol{\theta}_v, \sigma^2) \geq L(\boldsymbol{f}, \widehat{\boldsymbol{\theta}}_u, \widehat{\boldsymbol{\theta}}_v, \widehat{\sigma}^2). \tag{17}$$

Using the notation (12), (13), and (14), the likelihood of the data $\boldsymbol{f}$ can be written as

$$P(\boldsymbol{f} \,|\, \boldsymbol{\theta}_u, \boldsymbol{\theta}_v, \sigma^2) = (2\pi\sigma^2)^{-\frac{N+1}{2}} \exp\left(-\|\boldsymbol{f} - W\boldsymbol{\gamma}\|^2/(2\sigma^2)\right), \tag{18}$$

where $\|\cdot\|$ denotes the ℓ^2-Euclidean norm. Differentiating (18) with respect to σ^2 and setting the result to zero, we can obtain the MLE of σ^2:

$$\widehat{\sigma}^2 = \frac{1}{N+1}\|\boldsymbol{f} - W\boldsymbol{\gamma}\|^2. \tag{19}$$

Then, from the optimality of the Shannon code, the codelength of the data $\boldsymbol{f}$ given those parameters is bounded from below by the following negative log-likelihood of (18):

$$\begin{aligned} L(\boldsymbol{f} \,|\, \boldsymbol{\theta}_u, \boldsymbol{\theta}_v, \widehat{\sigma}^2) &\geq -\log P(\boldsymbol{f} \,|\, \boldsymbol{\theta}_u, \boldsymbol{\theta}_v, \widehat{\sigma}^2) \\ &= \frac{N+1}{2}\log\left(\frac{2\pi\mathrm{e}}{N+1}\|\boldsymbol{f} - W\boldsymbol{\gamma}\|^2\right). \end{aligned} \tag{20}$$

Hence, the MLEs of $\boldsymbol{\theta}_u$ and $\boldsymbol{\theta}_v$ can be obtained by minimizing (20), which is equivalent to

$$(\widehat{\boldsymbol{\theta}}_u, \widehat{\boldsymbol{\theta}}_v) = \arg\min_{\boldsymbol{\theta}_u, \boldsymbol{\theta}_v} \|\boldsymbol{f} - W\boldsymbol{\gamma}\|^2 = \arg\min_{\boldsymbol{\alpha}, \boldsymbol{\beta}} \|\boldsymbol{f} - R\boldsymbol{\alpha} - S\boldsymbol{\beta}\|^2. \tag{21}$$

Differentiating this functional by α_k and β_{ν_ℓ}, the MLE $\widehat{\boldsymbol{\alpha}}$ and $\widehat{\beta}_{\nu_\ell}$ must satisfy

$$R^T R\widehat{\boldsymbol{\alpha}} = R^T\left(\boldsymbol{f} - \sum_{\ell=1}^{M} \widehat{\beta}_{\nu_\ell}\boldsymbol{s}_{\nu_\ell}\right), \tag{22}$$

$$\widehat{\beta}_{\nu_\ell} = \boldsymbol{s}_{\nu_\ell}^T(\boldsymbol{f} - R\widehat{\boldsymbol{\alpha}}), \tag{23}$$

where $\boldsymbol{s}_{\nu_\ell} \in \mathbb{R}^{N+1}$ is the $(\nu_\ell + 1)$st column vector of S in (10). Eliminating $\widehat{\beta}_{\nu_\ell}$ in (22) using (23) with a bit of algebra, we obtain the following equation for $\widehat{\boldsymbol{\alpha}}$:

$$\widehat{\boldsymbol{\alpha}} = \left[R^T J_{N+1,M} R\right]^{-1}\left[R^T J_{N+1,M}\right]\boldsymbol{f}, \tag{24}$$

where

$$J_{N+1,M} \stackrel{\Delta}{=} I_{N+1} - \sum_{\ell=1}^{M} \boldsymbol{s}_{\nu_\ell} \boldsymbol{s}_{\nu_\ell}^T. \tag{25}$$

In practice, however, computing (24) and (25) gets complicated since we do not know a priori which M sinusoids out of $N-1$ sinusoids should be used to represent and approximate $\boldsymbol{v}$ component. In fact, if we want to minimize the functional (21) and obtain the MLEs $\widehat{\boldsymbol{\theta}}_u$ and $\widehat{\boldsymbol{\theta}}_v$, then we have to evaluate (24) and consequently (23) and (21) for each possible combination of M sinusoids $\{\boldsymbol{s}_{\nu_1}, \ldots, \boldsymbol{s}_{\nu_M}\}$ over $M = 0, \ldots, N-1$ and find the best one. Unfortunately, there are 2^{N-1} possible combinations, which is impractical even for moderate N. There are two possible approaches to circumvent this problem although both of them are suboptimal:

1) Restrict the M sinusoids to those of the *lowest* M frequencies, i.e., $\nu_\ell = \ell$, $\ell = 1, \ldots, M$. This approach still requires to compute (24) and (25), but we can certainly avoid the combinatorial explosion.
2) Force $\widehat{\boldsymbol{\alpha}} = (\boldsymbol{f}[0], \boldsymbol{f}[N] - \boldsymbol{f}[0])^T$ as if there is no noise on the boundary points and the noise exists only on the internal samples.

A. AMDL assuming M lowest frequency sinusoids (AMDL1)

In the first case, we use the M lowest frequency sinusoids to represent the $\boldsymbol{v}$ component. Thus, we can set $\nu_\ell = \ell$, $\ell = 1, \ldots, M$ in (23) and (25), which become

$$\widehat{\beta}_\ell = \boldsymbol{s}_\ell^T (\boldsymbol{f} - R\widehat{\boldsymbol{\alpha}}) \tag{26}$$

and

$$J_{N+1,M} \stackrel{\Delta}{=} I_{N+1} - \sum_{\ell=1}^{M} \boldsymbol{s}_\ell \boldsymbol{s}_\ell^T, \tag{27}$$

respectively. Using (27), we can compute (24) and obtain the best estimate for $\boldsymbol{\alpha}$ and consequently the best estimate for $\boldsymbol{\beta}$ via (26). With this information we can finally determine the codelength for $\boldsymbol{f}$ as in (16), which is

$$\begin{aligned} L(\boldsymbol{f}, \boldsymbol{\theta}_u, \boldsymbol{\theta}_v, \sigma^2) &= L(\sigma^2) + L(\boldsymbol{\theta}_u, \boldsymbol{\theta}_v \,|\, \sigma^2) + L(\boldsymbol{f} \,|\, \boldsymbol{\theta}_u, \boldsymbol{\theta}_v, \sigma^2) \\ &\geq L(\widehat{\sigma}^2) + L(\widehat{\boldsymbol{\theta}}_u, \widehat{\boldsymbol{\theta}}_v \,|\, \widehat{\sigma}^2) + L(\boldsymbol{f} \,|\, \widehat{\boldsymbol{\theta}}_u, \widehat{\boldsymbol{\theta}}_v, \widehat{\sigma}^2) \qquad (28) \\ &\stackrel{\Delta}{=} AMDL1(M). \qquad (29) \end{aligned}$$

We call the resulting description length (29), the "analytical MDL with lowest frequency sinusoids" and denote it by $AMDL1(M)$ where M refers to the use of the M sinusoids. To determine precisely each

term in (28) we will use our results from (19) and (20). The first term $L(\widehat{\sigma}^2)$ represents the description length for the estimated noise variance (in bits). Since $\widehat{\sigma}^2$ is a real-valued parameter, its description length is

$$L(\widehat{\sigma}^2) = \frac{1}{2}\log(N+1). \tag{30}$$

This is because for each real-valued parameter we assign the bit cost $(1/2)\log$(# data samples), which is $(1/2)\log(N+1)$ in this case, and whose asymptotic optimality was shown by Rissanen (see e.g., [8, Chap. 3]).

The second term in (28), $L(\widehat{\boldsymbol{\theta}}_u, \widehat{\boldsymbol{\theta}}_v \,|\, \widehat{\sigma}^2)$, represents the description length for the $\boldsymbol{u}$ and $\boldsymbol{v}$ components. Since $\widehat{\boldsymbol{\theta}}_u = (\widehat{\alpha}_0, \widehat{\alpha}_1)^T$ and $\widehat{\boldsymbol{\theta}}_v = (\widehat{\beta}_1, \ldots, \widehat{\beta}_M)^T$, we have $M+2$ real valued parameters. We also need to describe one integer parameter M that ranges between 0 and $N-1$, which requires $\log N$ bits. Thus the description length for this term becomes

$$L(\widehat{\boldsymbol{\theta}}_u, \widehat{\boldsymbol{\theta}}_v \,|\, \widehat{\sigma}^2) = \frac{M+2}{2}\log(N+1) + \log N. \tag{31}$$

Finally the last term in (28) is given by the last equality in (20). Summarizing all these terms, we have

$$\boxed{\begin{array}{l} AMDL1(M) = \\ \dfrac{M+3}{2}\log(N+1) + \dfrac{N+1}{2}\log(2\pi \mathrm{e}\cdot\widehat{\sigma}^2(M)) + \log N. \end{array}} \tag{32}$$

We now seek M over $0 \le M \le N-1$ that minimizes (32). Once we find the minimizer M^*, we can approximate the data $\boldsymbol{f}$ as

$$\boldsymbol{f} \approx R\widehat{\boldsymbol{\alpha}}^* + S\widehat{\boldsymbol{\beta}}^*, \tag{33}$$

where $\widehat{\boldsymbol{\alpha}}^*$ and $\widehat{\boldsymbol{\beta}}^*$ are the final MLEs using M^* in (24) and (26), respectively. The righthand side of (33) can be viewed as a denoised version of $\boldsymbol{f}$ whereas $(\widehat{\boldsymbol{\alpha}}^*, \widehat{\boldsymbol{\beta}}^*)$ can be viewed as its compressed representation.

B. AMDL assuming noiseless boundary points (AMDL2)

The endpoints $\boldsymbol{f}[0]$ and $\boldsymbol{f}[N]$ are now deterministic, so is the $\boldsymbol{u}$ component. Therefore there is no difference between $\boldsymbol{\alpha}$, $\boldsymbol{\theta}_u$, $\boldsymbol{u}$ and $\widehat{\boldsymbol{\alpha}}$, $\widehat{\boldsymbol{\theta}}_u$, $\widehat{\boldsymbol{u}}$, respectively. Note that we also reach to the same conclusion with $M = N-1$ even if we do not explicitly assume this no noise scenario at the endpoints. Let $\widetilde{\boldsymbol{f}} \stackrel{\Delta}{=} (\boldsymbol{f} - \boldsymbol{u})[1:N-1] = (\boldsymbol{f}[1] - \boldsymbol{u}[1], \cdots, \boldsymbol{f}[N-1] - \boldsymbol{u}[N-1])^T \in \mathbb{R}^{N-1}$, where $\boldsymbol{u} = R\boldsymbol{\alpha}$ as (7). Let $\widetilde{\boldsymbol{s}} = \boldsymbol{s}[1:N-1] \in \mathbb{R}^{N-1}$. Our simplified model thus has the following form instead of (6):

$$\widetilde{\boldsymbol{f}} = \widetilde{\boldsymbol{s}} + \widetilde{\boldsymbol{\eta}} = S_{N-1}\widetilde{\boldsymbol{\beta}} + \widetilde{\boldsymbol{\eta}} \in \mathbb{R}^{N-1}, \quad \widetilde{\boldsymbol{\eta}} \sim \mathcal{N}(\boldsymbol{0}, \sigma^2 I_{N-1}), \tag{34}$$

where $\widetilde{\boldsymbol{\beta}} \stackrel{\Delta}{=} \boldsymbol{\beta}[1:N-1]$, which has M nonzero entries. The description length (16) now becomes

$$\begin{aligned} L(\boldsymbol{f}, \boldsymbol{\theta}_u, \boldsymbol{\theta}_v, \sigma^2) &= L(\boldsymbol{\theta}_u) + L(\boldsymbol{\theta}_{\widetilde{v}}, \sigma^2) + L(\widetilde{\boldsymbol{f}} \,|\, \boldsymbol{\theta}_{\widetilde{v}}, \sigma^2) \\ &\geq L(\boldsymbol{\theta}_u) + L(\widehat{\boldsymbol{\theta}}_{\widetilde{v}}, \widehat{\sigma}^2) + L(\widetilde{\boldsymbol{f}} \,|\, \widehat{\boldsymbol{\theta}}_{\widetilde{v}}, \widehat{\sigma}^2) \qquad (35) \\ &\stackrel{\Delta}{=} AMDL2(M), \qquad (36) \end{aligned}$$

where $\boldsymbol{\theta}_{\widetilde{v}}$ is exactly the same as $\boldsymbol{\theta}_v$. In (36), we call the resulting description length as "analytical MDL with noiseless boundary", and denote it by $AMDL2(M)$ where M refers to the use of the M sinusoids. This functional $AMDL2$ of course depends also on $\widetilde{\boldsymbol{f}}$, $\widehat{\boldsymbol{\theta}}_{\widetilde{v}}$, $\widehat{\sigma}$, but we omit them in our notation. We now modify (18) accordingly as

$$P(\widetilde{\boldsymbol{f}} \,|\, \boldsymbol{\theta}_{\widetilde{v}}, \sigma^2) = (2\pi\sigma^2)^{-\frac{N-1}{2}} \exp\left(-\|\widetilde{\boldsymbol{f}} - S_{N-1}\widetilde{\boldsymbol{\beta}}\|^2/(2\sigma^2)\right). \qquad (37)$$

From this, (21), (19), (20) become

$$\widehat{\widetilde{\boldsymbol{\beta}}} = \arg \min_{\|\widetilde{\boldsymbol{\beta}}\|_0 = M} \|\widetilde{\boldsymbol{f}} - S_{N-1}\widetilde{\boldsymbol{\beta}}\|^2, \qquad (38)$$

$$\widehat{\sigma}^2 = \frac{1}{N-1}\|\widetilde{\boldsymbol{f}} - S_{N-1}\widehat{\widetilde{\boldsymbol{\beta}}}\|^2. \qquad (39)$$

$$L(\widetilde{\boldsymbol{f}} \,|\, \boldsymbol{\theta}_{\widetilde{v}}, \widehat{\sigma}^2) = \frac{N-1}{2}\log\left(2\pi \mathrm{e}\widehat{\sigma}^2\right). \qquad (40)$$

respectively, and we proceed our computation in this order. Note that $\|\widetilde{\boldsymbol{\beta}}\|_0 = M$ in (38) means that a set of vectors of length $N-1$ containing exactly M nonzero entries are searched for the minimum. Because $\|\widetilde{\boldsymbol{f}} - S_{N-1}\widetilde{\boldsymbol{\beta}}\| = \|S_{N-1}^T\widetilde{\boldsymbol{f}} - \widetilde{\boldsymbol{\beta}}\|$ thanks to the orthonormality of S_{N-1}, searching the minimum is now very easy: we can simply choose the sinusoids corresponding to the largest M coefficients of $S_{N-1}^T\widetilde{\boldsymbol{f}}$.

Now we can determine (36) precisely. The first term $L(\boldsymbol{\theta}_u)$ represents the description length (in bits) of the $\boldsymbol{u}$ component. Since $\boldsymbol{\theta}_u = (\alpha_0, \alpha_1)^T$, i.e., two real-valued parameters, we have

$$L(\boldsymbol{\theta}_u) = \log(N+1). \qquad (41)$$

The second term of (35) counts the description length of $\widehat{\boldsymbol{\theta}}_{\widetilde{v}}$ and the variance estimate $\widehat{\sigma}^2 = \widehat{\sigma}^2(M)$, which amounts to: 1) one integer parameter M that ranges between 0 and $N-1$; 2) $M+1$ real-valued parameters consisting of the M nonzero coefficients in $\widehat{\widetilde{\boldsymbol{\beta}}}$ and $\widehat{\sigma}^2$; and 3) M integer parameters, i.e., the indices $(\nu_1, \ldots, \nu_M)^T$, each of which ranges between 1 and $N-1$. Therefore, we have

$$L(\widehat{\boldsymbol{\theta}}_{\widetilde{v}}, \widehat{\sigma}^2) = \log N + \frac{M+1}{2}\log(N+1) + M\log(N-1). \qquad (42)$$

The last term in (42) can be further shortened as $1 + \min(M, N-1-M)\cdot\log(N-1)$, by recognizing that it is shorter to describe the indices of $N-1-M$ *zero* entries if $M \geq N/2$, provided that we add

the 1 bit flag to indicate whether the indices are those of zero entries or nonzero entries. Summarizing all these terms, we have

$$\boxed{\begin{aligned} AMDL2(M) = \frac{M+3}{2}\log(N+1) \\ + \min(M, N-1-M)\log(N-1) \\ + \frac{N-1}{2}\log\left(2\pi \mathrm{e}\cdot\widehat{\sigma}^2(M)\right) + \log N + 1. \end{aligned}} \tag{43}$$

We now seek M over $0 \le M \le N-1$ that minimizes (43). After finding the minimizer M^*, we obtain the approximation to the data $\boldsymbol{f}$ as

$$\boldsymbol{f} \approx R\boldsymbol{\alpha} + S\widehat{\boldsymbol{\beta}}^*,$$

which is different from (33) because $\boldsymbol{\alpha}$ is deterministic.

Remark IV.1. Theoretically, the minimizer M^* of the MDL cost (32) or (43) should be searched over all possible range of M, i.e., $0 \le M \le N-1$. However, in practice, when M reaches very close to $N-1$, the cost functions (32) and (43) are completely dominated by the fidelity term rather than the regularization (or model cost) term. Consequently, $M^* = N-1$ would be chosen, and we could achieve neither data compression nor noise removal. Hence, we limit the search range of M as $0 \le M \le C\cdot(N-1)$, where $0 < C < 1$ is typically chosen to be 0.6, i.e., slightly more than half the number of the DST coefficients.

V. Simultaneous Compression and Denoising with Quantization

A possibly better way to approach our simultaneous compression and denoising problem is to use a *scalar quantization* procedure to truly convert all the real-valued coefficients and parameters to integers by truncating them with some precision δ (which is to be optimized). In other words, we seek the shortest bitstream that can be stored as an actual *file* and from which we can recover a good approximation of the true signal with less noise.

If we want the quantized version of AMDL1, i.e., the model using the *lowest M frequency sinusoids*, then we need to compute (24) and (26) for each M and to quantize all the relevant parameters. This computational process is expensive when we apply it to our hierarchical segmentation that will be discussed in Section VI and thus is not practical for use.

So instead, let us assume that our simplified model (34) for our data $\boldsymbol{f}$ still holds true except one difference: we do not explicitly assume that the DST coefficient vector $\widetilde{\boldsymbol{\beta}}$ in (34) consists of M nonzero entries and the $N-1-M$ zeros. In our new formulation, the number of nonzero entries and that of zero

entries are completely and automatically controlled by δ. Thus instead of (36) we have the following total description length:

$$\begin{aligned} L(\boldsymbol{f}, \boldsymbol{\theta}_u, \boldsymbol{\theta}_v, \sigma^2, \delta) &= L(\delta) + L(\boldsymbol{\theta}_u \,|\, \delta) + L(\boldsymbol{\theta}_{\widetilde{v}}, \sigma^2 \,|\, \delta) \\ &\quad + L(\widetilde{\boldsymbol{f}} \,|\, \boldsymbol{\theta}_{\widetilde{v}}, \sigma^2, \delta) \\ &\geq L(\delta) + L(\boldsymbol{\theta}_u \,|\, \delta) + L(\widehat{\boldsymbol{\theta}}_{\widetilde{v}}, \widehat{\sigma}^2 \,|\, \delta) \\ &\quad + L(\widetilde{\boldsymbol{f}} \,|\, \widehat{\boldsymbol{\theta}}_{\widetilde{v}}, \widehat{\sigma}^2, \delta) \\ &\triangleq QMDL(\delta), \end{aligned} \tag{44}$$

where QMDL stands for "quantized MDL" and it depends on the parameter δ.

Let us analyze (44) so that we can determine an explicit codelength formula. In the first term of (44), the precision δ is encoded using

$$L(\delta) = \log(1/\delta) \tag{45}$$

bits. It is sometimes convenient to use the precision of the form $\delta = 2^{-q}$, $q \in \mathbb{N}$, which leads to $L(\delta) = q$ bits. Note that this is the key parameter in the minimization of (44) and thus it needs to be optimized.

The second term of (44) is the parameter vector $\boldsymbol{\alpha}$, which will be truncated with precision δ. For example, the parameter α_0 is approximated by $[\alpha_0/\delta] \cdot \delta$, where $[\cdot]$ is the nearest integer of its argument. Since δ is already recorded in the first term, we only need to store the integer $[\alpha_i/\delta]$ for α_i, $i = 0, 1$. Thus, we have

$$L(\boldsymbol{\theta}_u \,|\, \delta) = L^*\left([\alpha_0/\delta]\right) + L^*\left([\alpha_1/\delta]\right), \tag{46}$$

where $L^*(\cdot)$ is the codelength derived from the so-called *universal prior for integers* (see e.g., [8, Chap. 3]), which assigns the codelength for any integer $j \in \mathbb{Z}$ as follows:

$$L^*(j) = \begin{cases} 1 & \text{if } j = 0, \\ \log^* |j| + \log 4c_0 & \text{otherwise}, \end{cases} \tag{47}$$

where $\log^* |j|$ is the sum of iterated logarithms with only positive terms:

$$\log^* |j| = \log |j| + \log\log |j| + \cdots = \sum_{k>0} \max\left(\log^{(k)} |j|, 0\right),$$

where $\log^{(k)}(\cdot)$ is the k-times iterated logarithm. The constant $c_0 \approx 2.865064$ in (47) was derived so that equality holds in the Kraft inequality: $\sum_{j=-\infty}^{\infty} 2^{-L^*(j)} \leq 1$. Note that these truncated version of $\boldsymbol{\alpha}$ should be used to compute $\boldsymbol{u}$ and consequently $\boldsymbol{v}$ and the other quantities.

In the third term of (44), we quantize the entries of $\widehat{\boldsymbol{\theta}}_{\widetilde{v}} = \widehat{\widetilde{\boldsymbol{\beta}}} \in \mathbb{R}^{N-1}$ with precision δ. For simplicity, we adopt the so-called uniform quantization with "deadzone". The entire range of the coefficient values

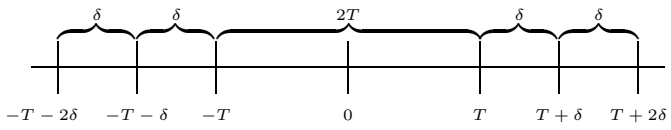


Fig. 1. The real line is subdivided into three major regions $(-\infty, -T) \cup [-T, T) \cup [T, \infty)$ where $(-\infty, -T)$ and (T, ∞) are further subdivided into regions of equal width δ.

is divided into a set of regions $(-\infty, -T) \cup [-T, T) \cup [T, \infty)$ where $T > 0$ and the regions except the "deadzone" $[-T, T)$ are further divided into a set of bins of the equal width δ as shown in Figure 1. The coefficients falling into a specific bin are replaced by the representative value of that bin, which is called the reconstruction value. For simplicity, we use the value of the midpoint of that bin as the reconstruction value so that we do not have to explicitly record them, i.e., we can recover the reconstruction value of any bin from its *bin index*. Also, the coefficients whose values lie within the deadzone are truncated to 0. This "thresholding" operation clearly serves as a denoising operation. As for the choice of the value T, it would be best to use the optimal theoretical threshold value by considering the nature of the signal, the DST coefficients $\widetilde{\boldsymbol{\beta}}$, and the statistics of the noise. Such choice was used by Chang et al. for wavelets [9]. For the ease of implementation, however, we will defer the theoretical question of determining the optimal value of T. Instead we will choose the best T among the finite number of possible values:

$$\left\{ n\delta \;\middle|\; n = 1, \ldots, \left\lceil \frac{\max(|\widetilde{\boldsymbol{\beta}}|)}{\delta} \right\rceil \right\}. \tag{48}$$

With such choice of T and with the symmetric quantization bins around the origin, all of the coefficients can be mapped into $2K + 1$ bins for some $K \in \mathbb{N}$ that depends on δ and the coefficient range. Thus, the quantization procedure converts $\widetilde{\boldsymbol{\beta}} \in \mathbb{R}^{N-1}$ to an integer-valued vector $\boldsymbol{n} \in \{-K, \ldots, K\}^{N-1}$ of the bin indices.

We can further reduce the codelength by applying a lossless entropy coding technique (e.g., Huffman or arithmetic coder) to this vector $\boldsymbol{n}$. In this paper, we use the Huffman coder that can further convert $\boldsymbol{n}$ into a bitstream of *at most* $(N-1)(H(\boldsymbol{p}) + 1)$ bits [10, Chap. 3], where $H(\cdot)$ computes the Shannon entropy of the probability mass function (pmf), and $\boldsymbol{p} = (p_{-K}, \ldots, p_K)$ is the pmf of $\boldsymbol{n}$, i.e., $p_k = \#\{i \in \{1, \ldots, N-1\} \,|\, n_i = k\}/(N-1)$. Note that if we directly encode $\boldsymbol{n}$ without using any entropy coder, then $(N-1)\log(2K+1)$ bits is required, which is the worst case scenario.

Lastly, to encode $\widehat{\sigma}^2$, we first compute (39) with the quantized DST coefficients. Then the codelength

of $\widehat{\sigma}^2$ with precision δ is $L^*([\widehat{\sigma}^2/\delta])$ bits. Hence, the third term of (44) can be approximated as

$$L(\widehat{\boldsymbol{\theta}}_{\widetilde{v}}, \widehat{\sigma}^2 \,|\, \delta) \approx L^*(K) + (N-1)(H(\boldsymbol{p}) + 1) + L^*([\widehat{\sigma}^2/\delta]). \tag{49}$$

Therefore, using (40), (45), (46), and (49), the total codelength (44) can be written as

$$\boxed{\begin{aligned} QMDL(\delta) = {} & \log(1/\delta) + L^*([\alpha_0/\delta]) + L^*([\alpha_1/\delta]) \\ & + L^*(K) + (N-1)(H(\boldsymbol{p}) + 1) \\ & + L^*([\widehat{\sigma}^2/\delta]) + \frac{N-1}{2} \log\left(2\pi \mathrm{e} \cdot \widehat{\sigma}^2\right), \end{aligned}} \tag{50}$$

where $\boldsymbol{p}$, $\widehat{\sigma}^2$, and K all depend on δ. We then search $\delta = \delta^*$ that minimizes (50) over a finite set of possible values. We will discuss an example of such a finite set in Section VII. Finally, with all the model parameters quantized with precision δ^* and encoded by the Huffman coder, we obtain the compressed bitstream representation of the denoised signal, which can be decoded and reconstructed at our disposal.

VI. Adaptive Hierarchical Segmentation, Compression, and Denoising

Based on our analysis of the global single segment case above, we now consider the hierarchical split of the input data and how to prune the tree-structured subintervals to obtain the best segmentation, which in turn should improve the compression and denoising performance as we discussed in Introduction. Let us assume that $N = 2^J$ for some $J \in \mathbb{N}$, and let us define a collection of the standard dyadic subintervals on the interval $[0,1]$, $\mathcal{I}_J \stackrel{\Delta}{=} \{I_{j,k} = [k/2^j, (k+1)/2^j] \,|\, j = 0, 1, \ldots, J-1,\ k = 0, 1, \ldots, 2^j - 1\}$. Let $N_j \stackrel{\Delta}{=} 2^{J-j}$. The number of available samples on $I_{j,k}$ including the two endpoints is $N_j + 1$ for all $k = 0, \ldots, 2^j - 1$ for a given level j, $0 \le j \le J-1$. Thus each of the shortest subintervals $I_{J-1,k}$ contains three samples and the longest interval $I_{0,0}$ contains the whole $2^J + 1$ samples.

We adopt the "split-and-merge" or "divide-and-conquer" approach à la best basis of Coifman and Wickerhauser [3]. In other words, we first split the input data into a collection of the data segments supported on $\mathcal{I}_J$, and at each node (or subinterval) $I_{j,k} \in \mathcal{I}_J$ we compute its MDL value by adjusting the formulas (32), (43), or (50) for $I_{j,k}$ instead of the whole interval $I = I_{0,0}$. Then we start the "merge" procedure by examining the bottom (finest) level nodes (i.e., $j = J-1$) whether they should be merged or not and continue this check from bottom to up until we reach to the root node. To determine whether two adjacent subintervals should be merged or not, we compare the MDL cost of the *union* of these two nodes with that of their parent node. If the cost of the union is smaller, we keep the children nodes; otherwise they are eliminated and we keep the parent node. Note, however, that our MDL cost functional

is *not additive*: we cannot simply add the MDL values of the children nodes already computed in the "split" stage. We also need to pay attention to the following:

- The midpoint of the parent node corresponds to the right endpoint (i.e., tail) of the left child node and the left endpoint (i.e., head) of the right child node. Consequently, when we compute the cost of the union of the children nodes, the MDL cost of this midpoint must be subtracted from the cost of the union.
- When we split the parent node into the left and right children nodes, we must add an additional two bits. This is because the cost of just representing the parent node is 1 bit (in the bit representation, it is 1 where 1 symbolizes the terminal node) and the cost of representing the two children nodes is 3 bits (in the bit representation, it is 011 where 0 symbolizes a split). Therefore, the difference is 2 bits. For a detailed explanation and examples see [5].

VII. Numerical Experiments

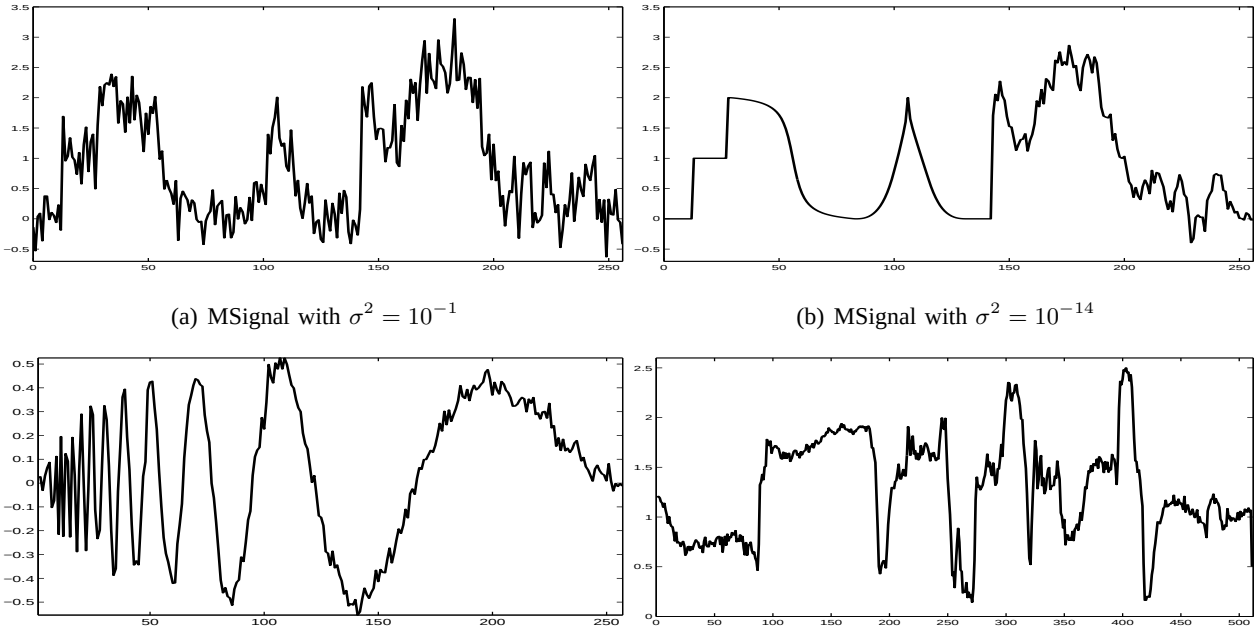
A. Experimental Data

To test the performance of our algorithms, we used four different datasets shown in Figure 2: a) a synthetic signal from Mallat's book [11, p. 81], which is referred to as "MSignal" with heavy AWGN of $\sigma^2 = 10^{-1}$; b) the MSignal with extremely weak AWGN of $\sigma^2 = 10^{-14}$; c) the "Doppler" signal available in the WaveLab software system [13] with moderate AWGN of $\sigma^2 = 10^{-5}$; and d) a single row from a standard digital image called "Peppers".

The MSignal has many interesting features: piecewise smooth components with several jump discontinuities in the signal values and derivatives in the first half and a noisy textured region in the last half; see Figure 2(b). The wavelet transforms are known to perform well on this signal thanks to the piecewise smooth nature of the first part of this signal. Note that the variances of AWGN are unknown to our algorithms.

For the Doppler signal, we added WGN with variance $\sigma^2 = 10^{-5}$ as shown in Figure 2(c). This is highly oscillatory, in particular, in the earlier part.

For the real dataset, we used a single row from a standard digital image as shown in Figure 2(d). More precisely, this is a normalized version of the 256th row of the standard image known as "Peppers". We assume that the real dataset has some amount of noise (of unknown variance).



(a) MSignal with $\sigma^2 = 10^{-1}$

(b) MSignal with $\sigma^2 = 10^{-14}$

(c) Doppler with $\sigma^2 = 10^{-5}$

(d) Real Signal (a row of the "Peppers" image)

Fig. 2. Plot of the signals used in our experiments.

B. Description of Experiments

Before presenting the results of our experiments, let us describe the types of experiments we conducted. Recall that in Sections IV and V we described two formulations for the PHLST-MDL method: 1) the "analytical" formulation (which has in turn two cases AMDL1 and AMDL2); and 2) the "quantized" formulation. We compared and assessed the performance of various algorithms by observing the compression ratio, relative ℓ^2-error, and MDL cost. In addition, we visually compared the signals reconstructed from their compressed representations.

For the experiments on the analytical MDL formulation, we used a *uniform precision* across all levels of decomposition, which means that the description length of each real-valued parameter for an input signal of length $N \in \mathbb{N}$ is fixed as $1/2 \cdot \log N$ and independent of the node in the tree.

For the experiments on the quantized MDL formulation, we also used a *uniform* δ, i.e., δ did not change across the levels while computing the MDL cost. Unlike the analytical formulation, however, for this formulation we searched the optimal $\delta = \delta^*$ from the set

$$\left\{ \frac{1}{\sqrt{2^{J-j}+1}} \,\middle|\, j = -2, -1, 0, \ldots, J-1 \right\}, \tag{51}$$

where $2^J + 1$ is the length of an input signal. Note that for each possible δ in (51), we also optimized

the threshold T over the finite set (48).

To compare the performance of our proposed methods objectively, we also implemented the WAVELET-MDL methods, which replace the PHLST representation by the wavelet representation and by applying the necessary modifications (e.g., the number of real-valued parameters, etc.) in our PHLST-MDL methods. In the WAVELET-MDL methods, we used the $D06$ wavelet transform (Daubechies 6-tap QMF with 3 vanishing moments, see [12]) using the WaveLab software [13]. In particular, we applied the $D06$ wavelet transform in three different ways:

1) Periodized Wavelet Transform (PWT);
2) Global line removal (similar to PHLST without any segmentation of the interval) followed by the Periodized Wavelet Transform (PWTLR);
3) Wavelets on the Interval (WOI); see [12, Chap. X].

The reason why we used these three different ways of applying the wavelet transform is that we wanted to see : 1) the effect of removing the linear structure connecting the head and tail of the signal to form a continuous periodic extension of a signal before applying the wavelet transform; and 2) the effect of the wavelet transform adjusted on the interval (i.e., WOI) without any such preprocessing.

Before describing the results of our experiments, we note several specifics about our algorithm setting. First, we decided to follow the noiseless boundary assumption for both the analytical (i.e., AMDL2) and quantized MDL formulations for the WAVELET-MDL methods. In other words, we did not examine the case corresponding to the PHLST-AMDL1 algorithm for the WAVELET-MDL methods. This is because the noise at boundary points are essentially immaterial for the WAVELET-MDL methods, which do not segment the input data explicitly in the time domain unlike the PHLST so that there are only two boundary points (the head and tail of the input data) for them. Second, our WAVELET-AMDL algorithms search the optimal number of wavelet coefficients to retain over 60% of the number of the sorted wavelet coefficients as described in Remark IV.1. Third, in order to apply these algorithms rapidly the length of an input signal should be $2^J + 1$ for some $J \in \mathbb{N}$ for the PHLST-MDL algorithms and 2^J for the WAVELET-MDL algorithms; hence, we cut one sample from the original signals when we applied the WAVELET-MDL algorithms. Fourth, the depth of the decomposition of each transform in the experiments was set to its deepest possible one: $J - 1$ for the PHLST; J for the PWT and PWTLR; and $J - 3$ for the WOI.

C. Results

In this subsection we present the results of our experiments by displaying a reconstruction plot and a numerical table for each formulation and for each noise level. In each reconstruction plot, the reconstructed signal is always plotted in a thick solid black line and overlays the original noiseless data plotted in a thin black line. The partitions (segmentations) obtained by the PHLST-MDL algorithms are shown as vertical lines. In each numerical table, we list *compression ratio*, *relative* ℓ^2*-error* (between the original noiseless signal and the reconstructed signal), and *MDL cost*. For each row of each table, the value in the *italic* font and the one in the ***bold italic*** font denote the worst and the best results among all the methods in that table, respectively.

We note that an ideal compression method would yield high compression ratio while maintaining the small relative ℓ^2-error. In any practical compression method, however, the higher the compression ratio, the larger the relative ℓ^2-error in general.

1) Mallat's Signal: We begin our evaluation of the results on MSignal with AWGN whose variance is $\sigma^2 = 10^{-1}$; see Table I. The first noticeable result for the analytical MDL experiments are the extreme values (best and worst result) for each category. The PHLST-AMDL1 produced the best relative ℓ^2-error and the worst compression ratio while the wavelets on the interval (WOI) produced the best compression ratio and the worst relative ℓ^2-error. The first three plots in Figure 3 show the signals reconstructed from the compressed representations by the PHLST-AMDL1, PHLST-AMDL2, and PWT-AMDL, respectively. The PHLST-AMDL1 produced a very smooth reconstruction tracing the original noiseless MSignal relatively closely except for the characteristic sharp features such as the step edges and the cusp. The PHLST-AMDL2 tried to follow those characteristic features more closely than the PHLST-AMDL1 by using the higher frequency sinusoids, but because of this, it also produced artifacts (e.g., the bump around the sample index $i = 70$). On the contrary, the PWT-AMDL (as well as the other two WAVELET-AMDL algorithms) produced much more visually annoying false sharp features due to the roughness of the $D06$ wavelet basis functions, which is particularly visible in the region before the textured region starts. Judging from the results in Table I and Figure 3, we conclude that the PHLST-AMDL1 produced the overall best result among the AMDL methods due to its low MDL cost, low relative ℓ^2-error, and overall visual quality of the reconstructed signal, with the expense of the compression ratio.

Let us now analyze the results of the quantized MDL methods. In Table II we see that each quantized MDL method produced better results than its analytical MDL counterpart. This is particularly so in the

TABLE I

ANALYTICAL MDL RESULTS ON MSIGNAL WITH AWGN WHOSE VARIANCE IS $\sigma^2 = 10^{-1}$.

	PHLST-AMDL1	PHLST-AMDL2	PWT	PWTLR	WOI
Compression ratio	*14.752*	17.350	19.932	17.246	***21.845***
Relative ℓ^2-error	***0.22219***	0.22920	0.27269	0.27375	*0.27457*
MDL cost	***264.97***	337.14	353.68	362.80	*366.53*

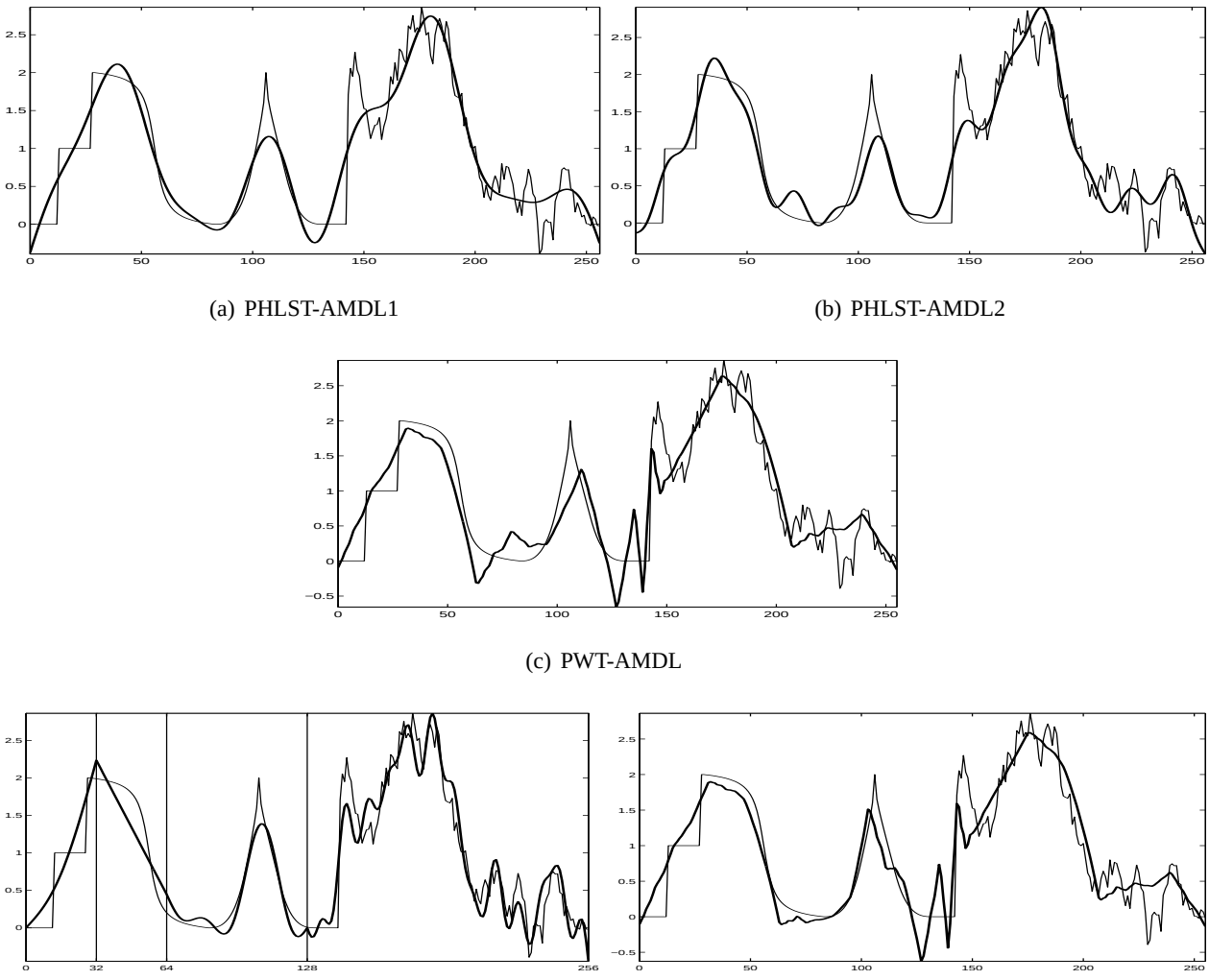
compression ratios, which is understandable because we store only integers after quantization instead of the double precision floating point numbers used in the analytical MDL methods. The best compression ratio and the worst relative ℓ^2-error was produced by the PWT while the worst compression ratio and the best relative ℓ^2-error was produced by the PHLST, which also produced the lowest MDL cost among all the QMDL methods. It is also interesting to note that the PHLST-QMDL method split the signal into four meaningful segments: 1) the step edges; 2) the smooth part; 3) the cusp region; and 4) the noisy textured region. We also list here the pairs $(j^*, n^*) \in \mathbb{N}^2$ that specify the optimal precision $\delta^* = 1/\sqrt{2^{8-j^*}+1}$ for PHLST and $\delta^* = 1/\sqrt{2^{8-j^*}}$ for wavelets, and the optimal threshold $T^* = n^*\delta^*$. These are $(6, 3)$, $(7, 2)$, $(7, 2)$, $(7, 2)$ for PHLST, PWT, PWTLR, and WOI, respectively. In other words, relatively large (i.e., low precision) δ^*'s were chosen for this heavy noise MSignal. Judging from the results in Tables I, II, and Figure 3, we conclude that the PHLST with QMDL formulation produced the best result among all the methods including those with the AMDL formulation for this highly noisy MSignal.

TABLE II

QUANTIZED MDL RESULTS ON MSIGNAL WITH AWGN WHOSE VARIANCE IS $\sigma^2 = 10^{-1}$.

	PHLST	PWT	PWTLR	WOI
Compression ratio	*102.33*	***117.99***	116.31	110.12
Relative ℓ^2-error	***0.20675***	*0.24787*	0.24787	0.22744
MDL cost	***300.72***	317.17	319.17	*324.77*

We now examine the results on MSignal with AWGN whose variance is $\sigma^2 = 10^{-14}$, i.e., almost noiseless case. Table III and the first three plots in Figure 4 show the results of the AMDL methods. The first thing we notice here is the drop of the compression ratios and the improvement of the relative ℓ^2-errors



(a) PHLST-AMDL1 (b) PHLST-AMDL2

(c) PWT-AMDL

(d) PHLST-QMDL (e) PWT-QMDL

Fig. 3. Signals reconstructed by our methods. The input signal is MSignal with AWGN whose variance is $\sigma^2 = 10^{-1}$. (a) The PHLST-AMDL1 method; (b) the PHLST-AMDL2 method; (c) the PWT-AMDL method that produced the lowest MDL cost among the wavelet-based methods; (d) the PHLST-QMDL method; (e) the PWT-QMDL method that produced the lowest MDL cost among the wavelet-based methods.

in all the methods. The PHLST-AMDL1 produced the lowest MDL cost with comparable compression ratio as that of the wavelet methods with moderate relative ℓ^2-error. A close examination of Figure 4(a) reveals, however, the undesired edge effect (overshoot and undershoot) around the step discontinuities around $i \in [32, 72]$. Moreover, it created an unintuitively large number of segmentations, particularly around the discontinuities. As for the overshoot and undershoot around the step discontinuities, our reasoning is as follows. If an original signal segment is a nonlinear curve without heavy noise, then

subtracting the MLE/least squares line computed by (24) with (27) in the PHLST-AMDL1 from that segment results in the nonzero boundary values in the v component. Since we use the DST basis vectors to approximate the v component, this essentially leads to a loss of the boundary points. Moreover, in the PHLST-AMDL1 reconstruction, the value at a common boundary point between two successive segments is computed by averaging the values of the u component of the left segment evaluated at that common boundary point and that of the right segment because the two least squares lines in the left and right segments do not match at the common boundary point in general. These contribute to the undesired edge effect around the step discontinuities. On the other hand, this does not happen in the PHLST-AMDL2 where the line passing through the two endpoints in each segment and the reconstruction value at the common boundary point between two successive segments is shared between them. As for the overly fine segmentations around the discontinuities in the PHLST-AMDL1, the following is our reasoning. The PHLST-AMDL1 generally does not provide us with a good approximation for the long subintervals particularly for the low noise case. This is because it only uses the M *lowest* frequency sinusoids instead of the best M frequency sinusoids, and moreover, as we mentioned in Remark IV.1, we restrict the search range of $M = M_j$ up to 60% of the number of sinusoids we can maximally have at level j, i.e., $0 \leq M_j \leq 0.6 \times 2^{J-j}$ to avoid the use of too many sinusoids. We found that the MDL value (or more precisely the AMDL1 value) of a node corresponding to a short subinterval in the piecewise constant region in MSignal is much smaller than that of its parent node due to the dominant fidelity term and the very small complexity term. Hence, this restriction in M in the AMDL1 formulation tends to produce finer segments, which is more severe than in the AMDL2 formulation. On the other hand, the PHLST-AMDL2 produced a quite reasonable and intuitive segmentation pattern and an excellent visually-pleasing reconstruction as can be seen in Figure 4(b). It did particularly nice job in discontinuous part, i.e., it produced progressively shorter segments toward the discontinuities. We also point out that the high frequency fluctuations in the textured part were considered as noise in the PHLST-AMDL methods. The WAVELET-AMDL methods resulted in the good relative ℓ^2-errors with the expense of the bad compression ratios. Figure 4(c) shows the reconstructed signal by the PWT-AMDL method, which yielded the lowest MDL cost among the WAVELET-AMDL methods. As we can see, the reconstruction is quite good over all except some Gibbs oscillations on the discontinuous steps in the earlier part and on the flat region before the large jump around the sample index $i = 150$ compared to the PHLST-AMDL2. Unlike the PHLST-AMDL models, the high frequency fluctuations in the textured part were considered as a part of the signal in the WAVELET-AMDL models, and hence they were not removed. We conclude that the PHLST-AMDL2 is the best among all the AMDL methods in terms of

its visually-pleasing reconstruction and the compression ratio.

TABLE III

ANALYTICAL MDL RESULTS ON MSIGNAL WITH AWGN WHOSE VARIANCE IS $\sigma^2 = 10^{-14}$.

	PHLST-AMDL1	PHLST-AMDL2	PWT	PWTLR	WOI
Compression ratio	*1.5104*	***2.1004***	1.5309	1.5128	1.5309
Relative ℓ^2-error	0.075521	*0.080101*	***0.0036657***	0.0037904	0.0054259
MDL cost	***-1668.5***	-1046.0	-16.050	4.2976	*57.795*

The quantized MDL results will be examined next. In Table IV, we again notice that the quantized MDL methods produced numerically better results than their analytical methods counterparts. The best compression ratio and the worst relative ℓ^2-error was achieved by the PHLST-QMDL.

The pairs $(j^*, n^*) \in \mathbb{N}^2$ for the precision δ^* and the threshold T^* are $(5, 1)$, $(-2, 1)$, $(-2, 1)$, $(-2, 1)$ for PHLST, PWT, PWTLR, and WOI, respectively. It is interesting to note that the WAVELET-QMDL methods all chose the very finest precision in the search range, i.e., $\delta^* = 1/\sqrt{2^{10}} \approx 0.03125$ while the PHLST-QMDL selected the coarser precision $\delta^* = 1/\sqrt{2^3 + 1} \approx 0.3333$. This is the reason why PHLST-QMDL got the highest compression ratio. Because we used the same quantization step for both the DST coefficients of the v components and the line parameters for the u components (or equivalently the two endpoints of the subintervals of the given input signal), the reconstructed signal of the PHLST-QMDL reveals the undesired kinks at the joints of the segments, in particular, the smooth region after the discontinuous steps around $i \in [32, 72]$. On the other hand, the partition generated by the PHLST-QMDL is quite reasonable and close to that of the PHLST-AMDL2 except in the smooth part around $i \in [64, 96]$.

Judging from the results in Table IV and Figure 4, it is hard to draw a conclusion for this MSignal with extremely low noise level. In terms of the compression ratio, the PHLST-QMDL is by far the best with the undesired artifacts in its reconstruction. In terms of the relative ℓ^2-error and the visual quality of the reconstruction, the WAVELET-QMDL methods, in particular, the PWT gives rise to the best except that it could not remove noise in the textured region.

2) The Doppler Signal: All of the tested AMDL methods had difficulty in capturing the fast oscillations in the beginning of the Doppler signal. The PHLST-AMDL1 yielded the smallest MDL cost, but the PHLST-AMDL2 produced the most visually-pleasing reconstruction among all the tested AMDL methods. The relative ℓ^2-error of the latter was also smaller than that of the the former, but it is slightly worse than that of the WAVELET-AMDL methods. This comes from the difficulty in capturing the fast oscillations

TABLE IV

QUANTIZED MDL RESULTS ON MSIGNAL WITH AWGN WHOSE VARIANCE IS $\sigma^2 = 10^{-14}$.

	PHLST	PWT	PWTLR	WOI
Compression ratio	***47.013***	15.501	15.443	*14.885*
Relative ℓ^2-error	*0.069999*	***0.0073826***	0.0074728	0.0082622
MDL cost	***-635.07***	-152.50	-144.06	*-61.370*

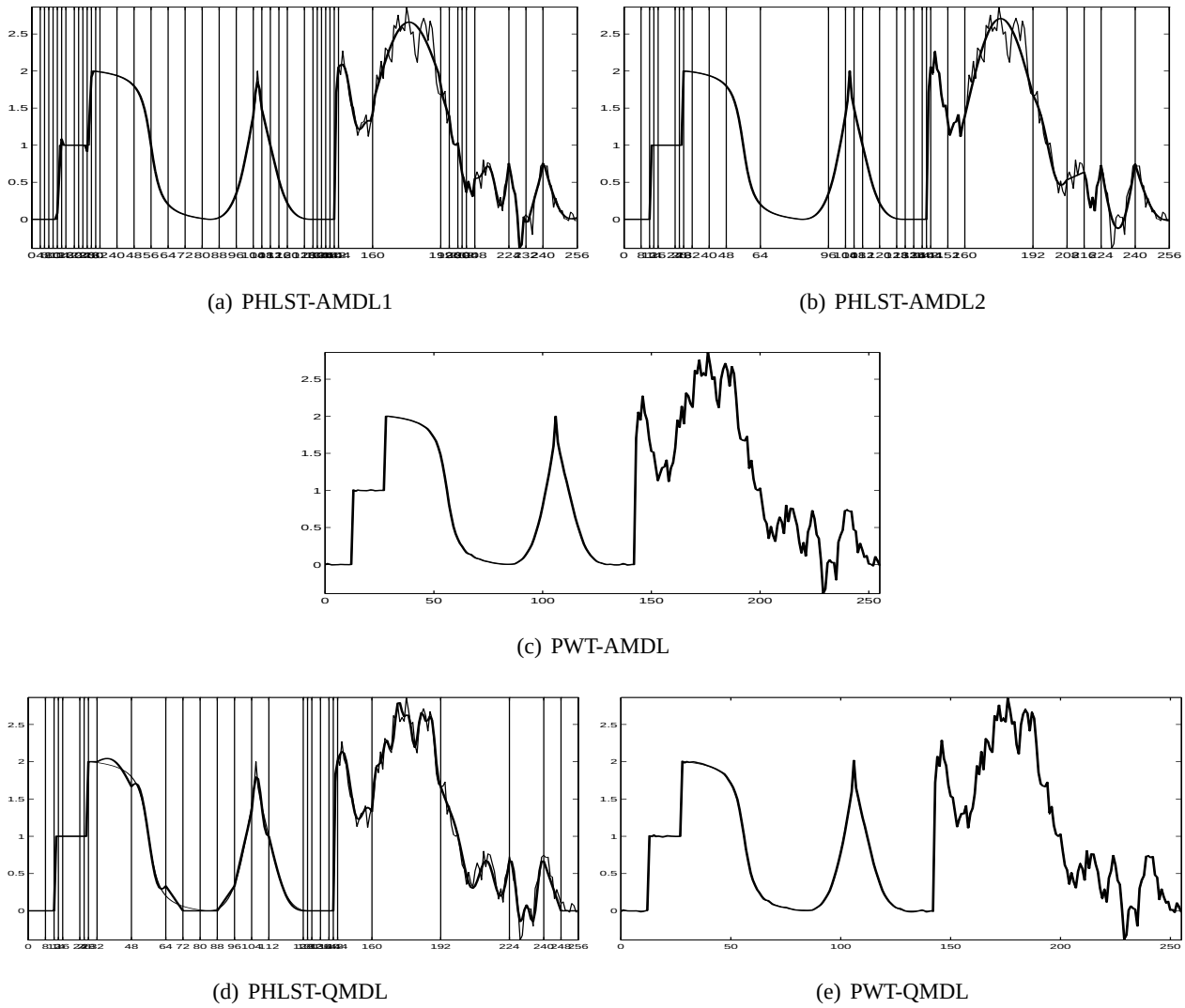
in the beginning although the WAVELET-AMDL methods could not capture that part well either. The partition pattern of the PHLST-AMDL2 is also much more reasonable than that of the PHLST-AMDL1. The former progressively becomes longer as the signal frequency decreases and is robust against noise except the first segment whereas the latter seems more sensitive to noise. In order to capture that fast oscillation, one needs to increase the search range of M over the coefficients for the minimization of the MDL cost. As we discussed in Remark IV.1, we set this range to 60% of the coefficients, which was not enough for capturing this fast oscillatory part. One cannot, however, increase this percentage too high since that would reduce the ability of the algorithms to compress and denoise the input data.

TABLE V

ANALYTICAL MDL RESULTS ON THE DOPPLER SIGNAL WITH AWGN WHOSE VARIANCE IS $\sigma^2 = 10^{-5}$.

	PHLST-AMDL1	PHLST-AMDL2	PWT	PWTLR	WOI
Compression ratio	*2.8169*	5.4771	6.0817	5.9578	***7.0197***
Relative ℓ^2-error	0.13275	0.13161	0.11400	***0.11338***	*0.14125*
MDL cost	***-587.59***	-385.20	-203.23	*-202.53*	-206.12

The results of the quantized MDL algorithms are far better than the analytical counterpart. The best result is obtained by the PHLST, and clearly this demonstrates the superiority of this method over the WAVELET-QMDL methods although the relative ℓ^2-error of the PHLST is slightly worse than those of the PWT and PWTLR but is better than that of the WOI. Interestingly, as shown in Figure 5, the QMDL methods could capture the fast oscillatory part unlike the AMDL methods. This suggests a potential superiority of the quantization process (in the QMDL methods) over the explicit coefficient selection (in the AMDL methods). We list here the pairs $(j^*, n^*) \in \mathbb{N}^2$ that specify the optimal quantization step and



(a) PHLST-AMDL1 (b) PHLST-AMDL2

(c) PWT-AMDL

(d) PHLST-QMDL (e) PWT-QMDL

Fig. 4. Signals reconstructed by our methods. The input signal is MSignal with AWGN whose variance is $\sigma^2 = 10^{-14}$. (a) The PHLST-AMDL1 method; (b) the PHLST-AMDL2 method; (c) the PWT-AMDL method that produced the lowest MDL cost among the WAVELET-AMDL methods; (d) the PHLST-QMDL method; (e) the PWT-QMDL method that produced the lowest MDL cost among the WAVELET-QMDL methods.

the threshold for each QMDL method: $(2, 2)$, $(2, 1)$, $(2, 1)$, $(3, 1)$, respectively. In other words, for this dataset, each method chose relatively similar parameters.

3) Real Dataset: The results of our experiments on the real dataset are summarized in Tables VII, VIII, and Figure 6. Among the AMDL methods, the PHLST-AMDL1 generated the smallest relative ℓ^2-error and the smallest MDL cost with the expense of the compression ratio. The WAVELET-PWT gave the best compression ratio with the expense of the relative ℓ^2-error. The deviation from the original

TABLE VI

QUANTIZED MDL RESULTS ON THE DOPPLER SIGNAL WITH AWGN WHOSE VARIANCE IS $\sigma^2 = 10^{-5}$.

	PHLST	PWT	PWTLR	WOI
Compression ratio	***64.326***	47.753	*47.476*	57.201
Relative ℓ^2-error	0.12347	***0.12074***	0.12074	*0.15161*
MDL cost	***-389.96***	-325.26	-323.26	*-290.83*

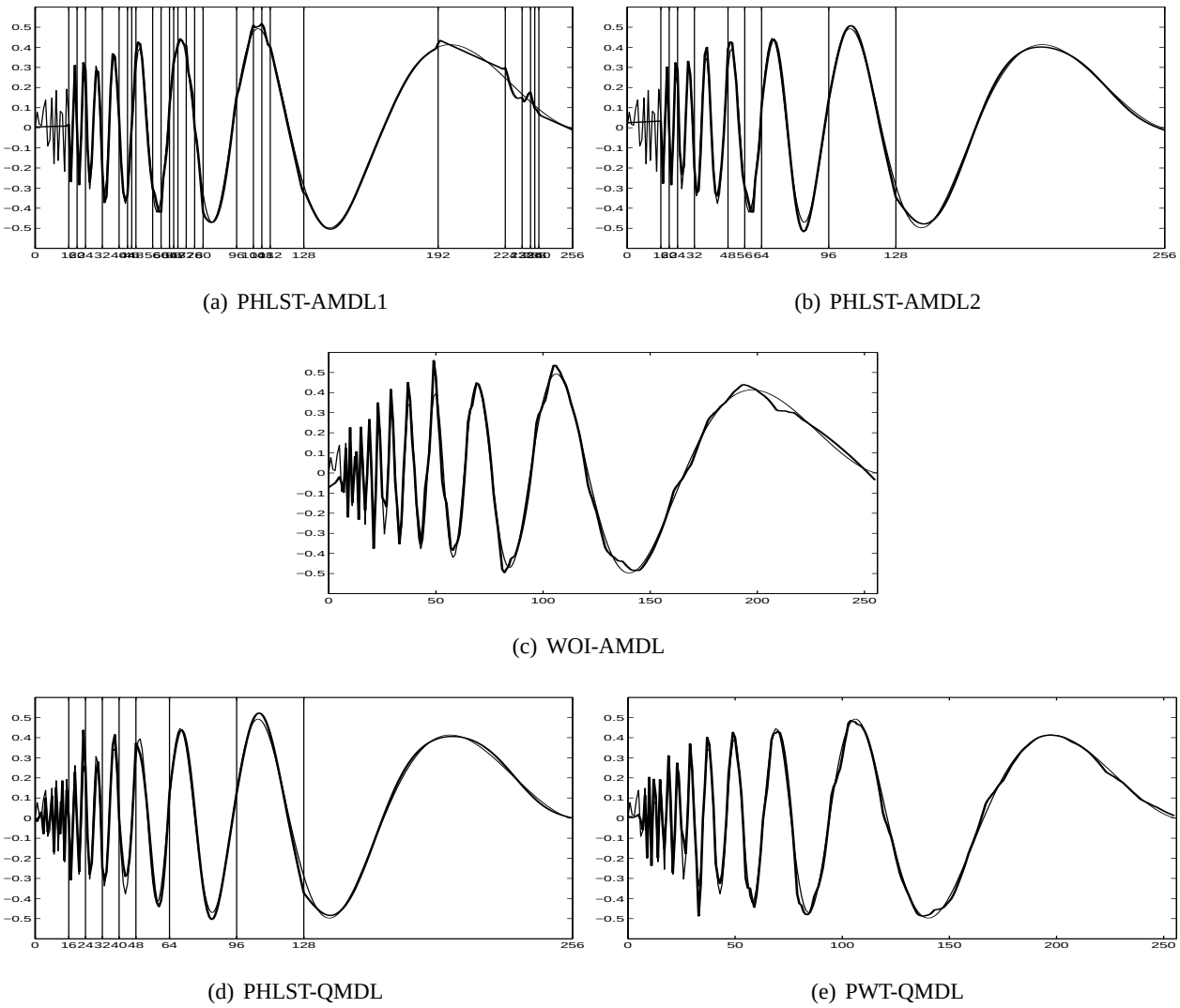
signal around the beginning and the end of the support was reduced in the WAVELET-PWTLR and the WAVELET-WOI compared to the WAVELET-PWT with the expense of the compression ratio and the MDL cost.

TABLE VII

ANALYTICAL MDL RESULTS ON THE REAL DATASET.

	PHLST-AMDL1	PHLST-AMDL2	PWT	PWTLR	WOI
Compression ratio	*3.4204*	5.0480	***8.5623***	7.5852	6.4657
Relative ℓ^2-error	***0.038769***	*0.059709*	0.058855	0.054261	0.055854
MDL cost	***-579.86***	-325.43	-85.082	-68.616	*106.57*

As for the QMDL methods, we again observe that they clearly performed better numerically than the AMDL methods. The best compression ratio was achieved by the PHLST method while the smallest relative ℓ^2-error was achieved by the PWTLR method. We also note that the reconstruction by the PHLST method decided to use the straight lines without the sinusoids in the interval around $[32, 80]$, which contributed to the best compression ratio and the worst relative ℓ^2-error. Overall, the PWTLR method seems to be the best choice for this signal, which is not surprising because this signal is quite suitable for the wavelet transform similarly to the MSignal case, and the global line removal helps the wavelet transform reduce the large size coefficients around the edges of the support interval. We again list the pairs $(j^*, n^*) \in \mathbb{N}^2$ that specify the optimal quantization step and the threshold for the QMDL methods: $(4, 1)$, $(3, 1)$, $(3, 1)$, $(3, 1)$, respectively.



(a) PHLST-AMDL1

(b) PHLST-AMDL2

(c) WOI-AMDL

(d) PHLST-QMDL

(e) PWT-QMDL

Fig. 5. Signals reconstructed by our methods. The input signal is the Doppler signal with AWGN whose variance is $\sigma^2 = 10^{-5}$. (a) The PHLST-AMDL1 method; (b) the PHLST-AMDL2 method; (c) the WOI-AMDL method that produced the lowest MDL cost among the wavelet-based methods; (d) the PHLST-QMDL method; and (e) the PWT-QMDL method that produced the lowest MDL cost among the wavelet-based methods.

VIII. Conclusion

We presented algorithms for simultaneous signal denoising, compression, and segmentation. In these algorithms we had two formulations: "analytical" and "quantized" formulations. The results of the experiments in Section VII showed the PHLST-MDL algorithms performing very well in compression, denoising, and segmentation of the observed noisy signal. In particular, the results on the oscillatory "Doppler" data were better than the WAVELET-MDL algorithms whereas the results on the piecewise

TABLE VIII

QUANTIZED MDL RESULTS ON THE REAL DATASET.

	PHLST	PWT	PWTLR	WOI
Compression ratio	***54.989***	48.365	46.558	*40.485*
Relative ℓ^2-error	*0.053274*	0.041854	***0.041053***	0.041577
MDL cost	-322.66	***-386.93***	-374.44	*-251.18*

smooth datasets (MSignal and the real dataset) were comparable qualitatively and quantitatively. We also observed that the "quantized" methods performed significantly better than the "analytical" methods. In particular, we observed lower relative ℓ^2-errors and higher compression ratios for the "quantized" MDL experiments in almost all cases since each parameter is quantized and converted into bit (or integer) representations. As for the computational cost of the PHLST-MDL algorithms, the expansion of a given input signal of length N into a full binary tree structured subspaces cost about $O(N[\log N]^2)$, which should be compared with $O(N)$ of the wavelet transforms.

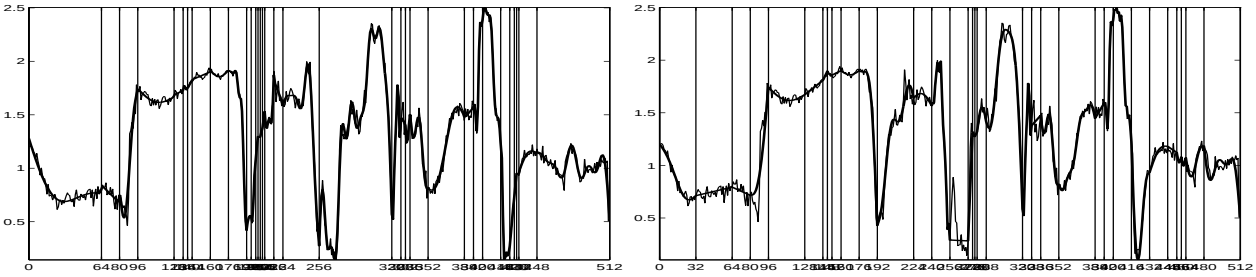
In order to improve our PHLST-MDL algorithms, we plan to investigate the following ideas:

- Estimation of boundary points using a local least squares method;
- A more elaborated search strategy for the optimal quantization precision rather than the simple minded strategy as (51).
- The use of different precision for encoding the $\boldsymbol{\theta}_u$ parameters from that for the $\boldsymbol{\theta}_v$ parameters; and
- The use of level dependent precisions and thresholds.

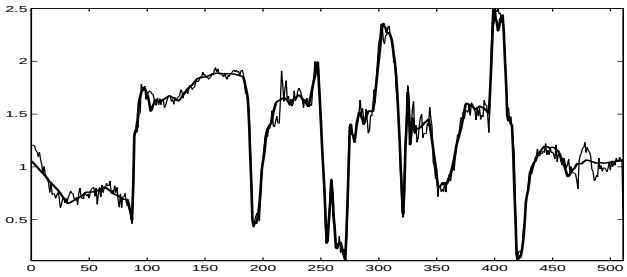
Furthermore, the most important advantage of the PHLST-MDL methods is their ability to perform interpolation, derivative estimation, and other feature computations *in the compressed representation* very easily thanks to formulas (1), (4), (8), and (9). On the contrary, in the wavelet-based representation, such computational tasks become much more involved and cumbersome. We hope to report this aspect of our algorithms at a later date.

Also as a future development, we plan to address the following important question: "Given a budget of B bits, what is the best PHLST representation of an input signal?"

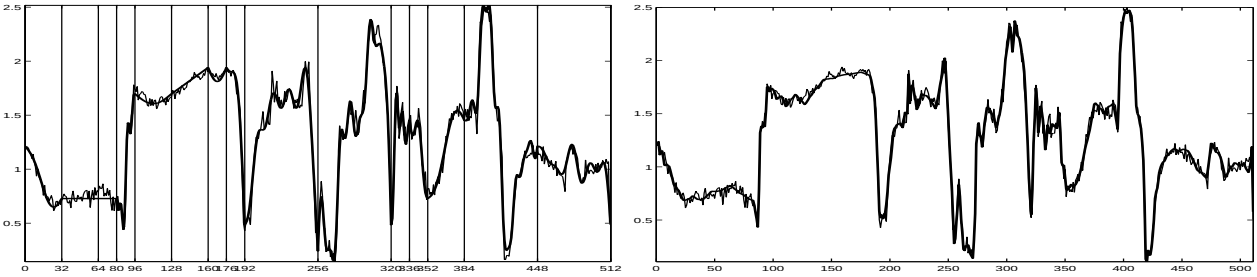
Finally we mention that it is straightforward to construct a 2D version of the PHLST-MDL algorithm for 2-D datasets.



(a) PHLST-AMDL1 (b) PHLST-AMDL2



(c) PWT-AMDL



(d) PHLST-QMDL (e) PWT-QMDL

Fig. 6. Signals reconstructed by our methods. The input signal is the 256th row of the standard image "peppers." (a) The PHLST-AMDL1 method; (b) the PHLST-AMDL2 method; (c) the PWT-AMDL method that produced the lowest MDL cost among the wavelet-based methods; (d) the PHLST-QMDL method; and (e) the PWT-QMDL method that produced the lowest MDL cost among the wavelet-based methods.

ACKNOWLEDGMENT

This research was partially supported by the ONR grants N00014-00-1-0469, N00014-06-1-0615, N00014-07-1-0166, and the NSF grant DMS-0410406. A preliminary version of a part of the material in this paper was presented at the 13th IEEE Workshop on Statistical Signal Processing, July 2005, Bordeaux, France [14].

REFERENCES

[1] N. Saito and J.-F. Remy, "A new local sine transform without overlaps: A combination of computational harmonic analysis and PDE," in *Wavelets: Applications in Signal and Image Processing X*, M. A. Unser, A. Aldroubi, and A. F. Laine, Eds., vol. Proc. SPIE 5207, 2003, pp. 495–506.

[2] ——, "The polyharmonic local sine transform: A new tool for local image analysis and synthesis without edge effect," *Applied and Computational Harmonic Analysis*, vol. 20, no. 1, pp. 41–73, 2006.

[3] R. R. Coifman and M. V. Wickerhauser, "Entropy-based algorithms for best basis selection," *IEEE Trans. Inform. Theory*, vol. 38, no. 2, pp. 713–719, Mar. 1992.

[4] N. Saito, "Simultaneous noise suppression and signal compression using a library of orthonormal bases and the minimum description length criterion," in *Wavelets in Geophysics*, E. Foufoula-Georgiou and P. Kumar, Eds. Academic Press, San Diego, CA, 1994, ch. XI, pp. 299–324.

[5] P. Moulin, "Signal estimation using adapted tree-structured bases and the MDL principle," in *Proc. IEEE-SP Intern. Symp. Time-Frequency and Time-Scale Analysis*, 1996, pp. 141–143.

[6] M. Hansen and B. Yu, "Wavelet thresholding via MDL: simultaneous denoising and compression," *IEEE Trans. Inform. Theory*, vol. 46, no. 5, pp. 1778–1788, 2000.

[7] C. Lanczos, *Discourse on Fourier Series*. New York: Hafner Publishing Co., 1966.

[8] J. Rissanen, *Stochastic Complexity in Statistical Inquiry*. Singapore: World Scientific, 1989.

[9] S. G. Chang, B. Yu, and M. Vetterli, "Adaptive wavelet thresholding for image denoising and compression," *IEEE Trans. Image Process.*, vol. 9, no. 9, pp. 1532–1546, 2000.

[10] K. Sayood, *Introduction to Data Compression*, 2nd ed. San Francisco, CA: Morgan Kaufmann Publishers, Inc., 2000.

[11] S. Mallat, *A Wavelet Tour of Signal Processing*, 2nd ed. San Diego, CA: Academic Press, 1999.

[12] I. Daubechies, *Ten Lectures on Wavelets*, ser. CBMS-NSF Regional Conference Series in Applied Mathematics. Philadelphia, PA: SIAM, 1992, vol. 61.

[13] D. Donoho *et al.*, http://www-stat.stanford.edu/~wavelab.

[14] N. Saito and E. R. Woei, "Simultaneous segmentation, compression, and denoising of signals using polyharmonic local sine transform and minimum description length criterion," in *Proc. 13th IEEE Workshop on Statistical Signal Processing*. IEEE, 2005, pp. 315–320.