

SIMULTANEOUS SEGMENTATION, COMPRESSION, AND DENOISING OF SIGNALS USING POLYHARMONIC LOCAL SINE TRANSFORM AND MINIMUM DESCRIPTION LENGTH CRITERION

Naoki Saito and Ernest Woei

Department of Mathematics
University of California, Davis
Davis, CA 95616

Email: saito@math.ucdavis.edu

ABSTRACT

We propose a new approach to simultaneously segment, compress, and denoise a given noisy signal by combining our compact signal representation scheme called *polyharmonic local sine transform* (PHLST) and the minimum description length (MDL) criterion. PHLST 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 (e.g., linear or cubic) 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 and without actually quantizing the relevant parameters. The resulting representation gives rise to simultaneous segmentation, compression, and denoising of the original data. We shall demonstrate its superiority over the best basis algorithm using the local cosine dictionary with the sparsity criterion.

1. 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) for images. However, the original PHLST assumes that the partition of an input signal is given a priori, and does not automatically compute the best possible partition of the signal. In this paper, we propose an automatic method to do that for 1D signals and give some 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) the signal into a set of local pieces brutally (i.e. by multiplying the characteristic functions) supported on dyadic subintervals in the form of a binary tree. Then, we “prune” this tree to come up with the “best” split or segmentation of the original signal. Here, the “best” means in terms of the *Minimum Description Length* (MDL) criterion. We shall

This work was partially supported by ONR YIP N00014-00-1-0469 and NSF DMS-0410406.

discuss two different versions of our basic formulation for simultaneous compression and denoising in Sections 4 and 5. Section 6 further develops our algorithm for signal segmentation, which will be followed by our numerical experiments in Section 7. But first, let us review briefly our PHLST scheme in Section 2 and set up our noisy signal model in Section 3.

2. REVIEW OF PHLST

We shall review the one-dimensional and global (i.e., without subdividing the domain) version 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 $m \geq 1$. Now, we split the data function into two pieces

$$f(x) = u(x) + v(x). \quad (1)$$

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

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

with the boundary condition

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

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

$$u(x) = \alpha_0 + \alpha_1 x, \quad (4)$$

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

Remark 2.1. 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., [4, Sec. 19]), which we are currently investigating, we shall 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. Note also that for higher dimensions, the solution of the polyharmonic equation 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

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

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

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

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 the ordinary Fourier series expansion with the periodic condition giving rise to $O(\ell^{-1})$ with the infamous Gibbs phenomenon [4, Sec. 10] or the Fourier cosine series expansion with the Neumann boundary condition giving 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 fast decaying 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, i.e., α_k 's in (4). 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 the trigonometric polynomials to approximate the data, we would encounter the Gibbs phenomenon. On the other hand, if we were to use only the algebraic polynomial (of high degree) to approximate the data, then we would encounter with the so-called Runge phenomenon [4, 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.

3. OUR SIGNAL MODEL

Let us now consider the case where the data contain white Gaussian noise with unknown variance σ^2 . Suppose the data are sampled uniformly at $x_n = n/N$, $n = 0, 1, \dots, N$. Thus, our signal model can be written as

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

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

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

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

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

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

We assume that the v component consists of M sinusoids ($0 \leq M \leq N-1$) with frequencies $1 \leq \nu_1, \dots, \nu_M \leq N-1$ instead of $N-1$ sinusoids of frequencies $1, \dots, N-1$:

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

where

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

These are a subset of the Discrete Sine Transform (in fact the so-called DST Type I) coefficients of the v component. The reason why we model the v component by M sinusoids instead of $N-1$ sinusoids is the following. The column vectors of U in (7) and $N-1$ 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. Let us write the v component as

$$\mathbf{v} = V\boldsymbol{\beta}, \quad V \triangleq \begin{bmatrix} 0 & \cdots & 0 \\ \vdots & V_{N-1} & \vdots \\ 0 & \cdots & 0 \end{bmatrix} \in \mathbb{R}^{(N+1) \times (N+1)}, \quad (8)$$

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

$$\mathbf{f} = U\boldsymbol{\alpha} + V\boldsymbol{\beta} + \boldsymbol{\eta}, \quad \boldsymbol{\eta} \sim \mathcal{N}(\mathbf{0}, \sigma^2 I_{N+1}), \quad (9)$$

which can be further simplified as

$$\mathbf{f} = W\boldsymbol{\gamma} + \boldsymbol{\eta}, \quad \boldsymbol{\eta} \sim \mathcal{N}(\mathbf{0}, \sigma^2 I_{N+1}) \quad (10)$$

by defining

$$W \triangleq [U(:, 1) | V(:, 2:N) | U(:, 2)] \in \mathbb{R}^{(N+1) \times (N+1)}, \quad (11)$$

$$\boldsymbol{\gamma} \triangleq \begin{bmatrix} \alpha_0 \\ \boldsymbol{\beta}[1:N-1] \\ \alpha_1 \end{bmatrix} \in \mathbb{R}^{N+1}. \quad (12)$$

Note that the matrix W is *not orthogonal*.

4. ANALYTICAL COMPRESSION AND DENOISING

The essence of MDL is the following. Suppose that we are given data $\mathbf{d} \in \mathbb{R}^n$ that were generated by some parametric statistical model $P(\mathbf{d} | \boldsymbol{\theta})$ 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., [5]) that the best model is the minimizer of the following cost functional (or codelength):

$$L(\mathbf{d}, \boldsymbol{\theta}) = L(\mathbf{d} | \boldsymbol{\theta}) + L(\boldsymbol{\theta}) \approx -\log P(\mathbf{d} | \boldsymbol{\theta}) + \frac{k}{2} \log n. \quad (13)$$

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 the MDL criterion in our problem. One is called the ‘‘analytical’’ formulation, the other is the ‘‘quantized’’ formulation (see [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 the model parameters and to compute their maximum likelihood estimates (MLEs). We also used this strategy in our earlier paper [7] for signal compression and denoising using the wavelet packets and local trigonometric dictionaries. On the other hand, the quantized formulation, which we shall discuss in Section 5, actually performs the quantization of all the parameters, truly converts everything into ‘‘bits’’, and seeks the model that generates the shortest bitstream for given data.

Let θ_u and θ_v be the vectors of parameters that completely specify the \mathbf{u} and \mathbf{v} components in (7) and (8), respectively. It is clear that θ_u is simply the pair (m, α) if we have a choice in m , say $m = 1$ or 2 . Since we only consider the $m = 1$ case, we can assume that $\theta_u = \alpha \in \mathbb{R}^2$. As for θ_v , it is a concatenation of M nonzero real-valued DST coefficients $(\beta_{\nu_1}, \dots, \beta_{\nu_M})^T \in \mathbb{R}^M$ and their indicator vector, $(\nu_1, \dots, \nu_M)^T \in \{1, \dots, N-1\}^M$. Therefore, the codelength (13) of our data with our signal model (6), (9), (10), can be written as

$$L(\mathbf{f}, \theta_u, \theta_v, \sigma^2) = L(\sigma^2) + L(\theta_u, \theta_v | \sigma^2) + L(\mathbf{f} | \theta_u, \theta_v, \sigma^2) \quad (14)$$

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

$$L(\mathbf{f}, \theta_u, \theta_v, \sigma^2) \geq L(\mathbf{f}, \hat{\theta}_u, \hat{\theta}_v, \hat{\sigma}^2). \quad (15)$$

Using the notation (10), (11), (12), the likelihood of the data \mathbf{f} can be written as

$$P(\mathbf{f} | \theta_u, \theta_v, \sigma^2) = (2\pi\sigma^2)^{-\frac{N+1}{2}} \exp(-\|\mathbf{f} - W\gamma\|^2 / (2\sigma^2)). \quad (16)$$

Differentiating (16) with respect to σ^2 and setting the result to zero, we can obtain the MLE of σ^2 :

$$\hat{\sigma}^2 = \frac{1}{N+1} \|\mathbf{f} - W\gamma\|^2. \quad (17)$$

Then, from the optimality of the Shannon code, the codelength of the data \mathbf{f} given those parameters is bounded from below by the following negative log-likelihood of (16):

$$\begin{aligned} L(\mathbf{f} | \theta_u, \theta_v, \hat{\sigma}^2) &\geq -\log P(\mathbf{f} | \theta_u, \theta_v, \hat{\sigma}^2) \\ &= \frac{N+1}{2} \log \left(\frac{2\pi e}{N+1} \|\mathbf{f} - W\gamma\|^2 \right). \end{aligned} \quad (18)$$

Hence, the MLEs of θ_u and θ_v can be obtained by minimizing (18), which is equivalent to

$$(\hat{\theta}_u, \hat{\theta}_v) = \arg \min_{\theta_u, \theta_v} \|\mathbf{f} - W\gamma\|^2 = \arg \min_{\theta_u, \theta_v} \|\mathbf{f} - U\alpha - V\beta\|^2. \quad (19)$$

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

$$U^T U \hat{\alpha} = U^T \left(\mathbf{f} - \sum_{\ell=1}^M \hat{\beta}_{\nu_\ell} \mathbf{v}_{\nu_\ell} \right), \quad (20)$$

$$\hat{\beta}_{\nu_\ell} = \mathbf{v}_{\nu_\ell}^T (\mathbf{f} - V \hat{\alpha}), \quad (21)$$

where $\mathbf{v}_{\nu_\ell} \in \mathbb{R}^{N+1}$ is the ν_ℓ th column vector of V in (8). Eliminating $\hat{\beta}_{\nu_\ell}$ in (20) using (21) with a bit of algebra, we obtain the following equation for $\hat{\alpha}$:

$$\hat{\alpha} = [U^T J_{N+1, M} U]^{-1} [U^T J_{N+1, M}] \mathbf{f}, \quad (22)$$

where

$$J_{N+1, M} \triangleq I_{N+1} - \sum_{\ell=1}^M \mathbf{v}_{\nu_\ell} \mathbf{v}_{\nu_\ell}^T. \quad (23)$$

In practice, however, computing (22) and (23) gets complicated since we do not know a priori which M sinusoids out of $N-1$ sinusoids should be used to represent and approximate \mathbf{v} component. If we were to truly minimize the functional (19) and obtain the MLEs $\hat{\theta}_u$ and $\hat{\theta}_v$, then we would need to evaluate (22) and consequently (21) and (19) for each possible combination of M sinusoids $\{\mathbf{v}_{\nu_1}, \dots, \mathbf{v}_{\nu_M}\}$ over $M = 0, \dots, N-1$ and find the best one. Unfortunately, there are 2^{N-1} possible combinations. There are two possible approaches to circumvent this problem although both of them are suboptimal. One is to restrict the M sinusoids to those of the *lowest* M frequencies, i.e., $\nu_\ell = \ell$, $\ell = 1, \dots, M$. This approach still requires to compute (22) and (23), but we can certainly avoid the combinatorial explosion. The other approach is to force $\hat{\alpha} = (\mathbf{f}[0], \mathbf{f}[N] - \mathbf{f}[0])^T$, as if there is no noise on the boundary points and the noise exists only on the internal samples. We are currently experimenting and investigating the first approach and shall report our findings at a later date. In this paper, we shall focus on the second approach. The endpoints $\mathbf{f}[0]$ and $\mathbf{f}[N]$ are now deterministic, so is the \mathbf{u} component. Therefore there is no difference between α , θ_u , \mathbf{u} and $\hat{\alpha}$, $\hat{\theta}_u$, $\hat{\mathbf{u}}$, 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 $\tilde{\mathbf{f}} \triangleq (\mathbf{f} - \mathbf{u})[1 : N-1] = (\mathbf{f}[1] - \mathbf{u}[1], \dots, \mathbf{f}[N-1] - \mathbf{u}[N-1])^T \in \mathbb{R}^{N-1}$, where $\mathbf{u} = U\alpha$ as (7). Let $\tilde{\mathbf{v}} = \mathbf{v}[1 : N-1] \in \mathbb{R}^{N-1}$. Our simplified model thus has the following form instead of (6):

$$\tilde{\mathbf{f}} = \tilde{\mathbf{v}} + \tilde{\boldsymbol{\eta}} = V_{N-1} \tilde{\boldsymbol{\beta}} + \tilde{\boldsymbol{\eta}} \in \mathbb{R}^{N-1}, \quad \tilde{\boldsymbol{\eta}} \sim \mathcal{N}(\mathbf{0}, \sigma^2 I_{N-1}), \quad (24)$$

where $\tilde{\boldsymbol{\beta}} \triangleq \beta[1 : N-1]$, which has M nonzero entries. The description length (14) now becomes

$$\begin{aligned} L(\mathbf{f}, \theta_u, \theta_v, \sigma^2) &= L(\theta_u) + L(\theta_v, \sigma^2) + L(\tilde{\mathbf{f}} | \theta_v, \sigma^2) \\ &\geq L(\theta_u) + L(\hat{\theta}_v, \hat{\sigma}^2) + L(\tilde{\mathbf{f}} | \hat{\theta}_v, \hat{\sigma}^2) \\ &\triangleq \text{AMDL}(M), \end{aligned} \quad (25)$$

where θ_v is exactly the same as θ_v . In (25), we call the resulting description length as ‘‘analytical MDL’’ (AMDL), and denote by $\text{AMDL}(M)$ to signify its dependence on M . This AMDL of course depends also on $\tilde{\mathbf{f}}$, $\hat{\theta}_v$, $\hat{\sigma}^2$, but we omit them in our notation. We now modify (16) accordingly as

$$P(\tilde{\mathbf{f}} | \theta_v, \sigma^2) = (2\pi\sigma^2)^{-\frac{N-1}{2}} \exp\left(-\|\tilde{\mathbf{f}} - V_{N-1} \tilde{\boldsymbol{\beta}}\|^2 / (2\sigma^2)\right). \quad (26)$$

From this, (19), (17), (18) become

$$\hat{\tilde{\boldsymbol{\beta}}} = \arg \min_{\|\tilde{\boldsymbol{\beta}}\|_0 = M} \|\tilde{\mathbf{f}} - V_{N-1} \tilde{\boldsymbol{\beta}}\|^2, \quad (27)$$

$$\hat{\sigma}^2 = \frac{1}{N-1} \|\tilde{\mathbf{f}} - V_{N-1} \hat{\tilde{\boldsymbol{\beta}}}\|^2. \quad (28)$$

$$L(\tilde{\mathbf{f}} | \theta_{\tilde{v}}, \hat{\sigma}^2) = \frac{N-1}{2} \log \left(\frac{2\pi e}{N-1} \hat{\sigma}^2 \right). \quad (29)$$

respectively, and we proceed our computation in this order. Note that $\|\tilde{\beta}\|_0 = M$ in (27) means that a set of vectors of length $N-1$ containing only M nonzero entries are searched for the minimum. Because $\|\tilde{\mathbf{f}} - V_{N-1}\tilde{\beta}\| = \|V_{N-1}^T \tilde{\mathbf{f}} - \tilde{\beta}\|$ thanks to the orthonormality of V_{N-1} , searching the minimum is now very easy: we can simply choose the sinusoids corresponding to the largest M coefficients of $V_{N-1}^T \tilde{\mathbf{f}}$.

Finally let us determine precisely each term of $AMDL(M)$ in (25). The first term $L(\theta_u)$ represents the description length (in bits) of the u component. Since $\theta_u = (\alpha_0, \alpha_1)^T$, i.e., two real-valued parameters, we have

$$L(\theta_u) = \log(N+1). \quad (30)$$

Note that for each real-valued parameter we assign the cost $(1/2) \cdot \log(\# \text{ data samples})$, which is $(1/2) \log(N+1)$ in this case, and whose asymptotic optimality was shown by Rissanen (see e.g., [5, Chap. 3]). The second term of (25) counts the description length of $\hat{\theta}_{\tilde{v}}$ and the variance estimate $\hat{\sigma}^2 = \hat{\sigma}^2(M)$, which amounts to one integer parameter M that ranges between 0 and $N-1$, $M+1$ real-valued parameters consisting of the M nonzero coefficients in $\hat{\beta}$ and $\hat{\sigma}^2$, and $M+1$ integer parameters, i.e., the indices $(\nu_1, \dots, \nu_M)^T$, each of which ranges between 1 and $N-1$. Therefore, we have

$$L(\hat{\theta}_{\tilde{v}}, \hat{\sigma}^2) = \log N + \frac{M+1}{2} \log(N+1) + M \log(N-1). \quad (31)$$

The last term in (31) 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

$$\begin{aligned} AMDL(M) &= \frac{M+3}{2} \log(N+1) \\ &+ \min(M, N-1-M) \log(N-1) \\ &+ \frac{N-1}{2} \log(2\pi e \cdot \hat{\sigma}^2(M)) + \log N + 1. \end{aligned} \quad (32)$$

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

$$\mathbf{f} \approx U\alpha + V\hat{\beta},$$

which can be viewed as a denoised version of \mathbf{f} whereas $(\alpha, \hat{\beta})$ can be viewed as its compressed representation.

5. 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 without noise. Let us assume our

simplified model (24) for the given data is still true except one difference: we do not explicitly assume that the DST coefficient vector β in (24) consists of M nonzero entries and $N-1-M$ zeros. In our new formulation, the number of nonzero entries and that of zero entries are completely controlled by δ . Therefore, instead of (25), the total description length with quantization becomes

$$\begin{aligned} L(\mathbf{f}, \theta_u, \theta_v, \sigma^2, \delta) &= L(\delta) + L(\theta_u | \delta) + L(\theta_v, \sigma^2 | \delta) \\ &+ L(\tilde{\mathbf{f}} | \theta_{\tilde{v}}, \sigma^2, \delta) \\ &\geq L(\delta) + L(\theta_u | \delta) + L(\hat{\theta}_{\tilde{v}}, \hat{\sigma}^2 | \delta) \\ &+ L(\tilde{\mathbf{f}} | \hat{\theta}_{\tilde{v}}, \hat{\sigma}^2, \delta) \\ &\triangleq QMDL(\delta), \end{aligned} \quad (33)$$

where QMDL stands for ‘‘quantized MDL’’, and we signify its dependence on δ in our notation.

Let us analyze each term of (33). First, the precision δ , which is the key parameter to be optimized to obtain the minimum of (33), needs to be recorded and stored. In general, encoding δ requires $L(\delta) = \log(1/\delta)$ bits. Note that we often use the precision of the form $\delta = 2^{-q}$, $q \in \mathbb{N}$, so in this case, $L(\delta) = q$ bits.

In the second term of (33), the parameter vector α must now 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 $[\alpha_0/\delta]$ for α_0 . Thus, we have

$$L(\theta_u | \delta) = L^*([\alpha_0/\delta]) + L^*([\alpha_1/\delta]), \quad (34)$$

where $L^*(\cdot)$ is the codelength derived from the so-called *universal prior for integers* (see e.g., [5, Chap. 3]), which assigns the codelength for any integer $j \in \mathbb{Z}$ as follows:

$$L^*(j) = \begin{cases} 1 & \text{if } j = 0, \\ \log^* |j| + \log 4c_0 & \text{otherwise,} \end{cases} \quad (35)$$

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

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

where $\log^{(k)}(\cdot)$ is the k -times iterated logarithm. The constant $c_0 \approx 2.865064$ in (35) 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 α should be used to compute u .

Now, in the third term of (33), the entries of $\theta_{\tilde{v}} = \tilde{\beta} \in \mathbb{R}^{N-1}$ must be quantized with precision δ . For simplicity, we adopt the so-called uniform quantization with ‘‘deadzone’’. The entire range of the coefficient values 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 . The coefficients falling into a specific bin are replaced by the representative value of that bin called the reconstruction level. Again for simplicity, we use the value of the midpoint of that bin so that we do not have to explicitly record the reconstruction levels. Thus, note that the coefficients whose values are in the deadzone are truncated to 0, and this clearly serves as a threshold operation in denoising. Thanks to these simplification, we can recover the reconstruction level of any bin from its *bin index*. As for the choice of T , it would be the best to use the optimal

theoretical threshold value by considering the nature of the signal, the DST coefficients $\tilde{\beta}$, and the statistics of the noise as was done by Chang et al., [8] for wavelets, if such theoretical value is available. For the ease of implementation, however, we shall adopt a very simple strategy at this point and defer the theoretical question for the future study: we set $T = \delta$, i.e., the deadzone is twice as wide as the other bins. Now, the entire coefficients can be mapped into $2K + 1$ bins for some $K \in \mathbb{N}$ that depends on δ and the coefficient range. Thus, our quantization process converts $\tilde{\beta} \in \mathbb{R}^{N-1}$ to an integer-valued vector $\mathbf{n} \in \{-K, \dots, K\}^{N-1}$ of the bin indices. Any lossless entropy coding technique, e.g., the Huffman or arithmetic coder, should be used to store this integer vector \mathbf{n} efficiently. In this paper, we use the Huffman coder that can further convert \mathbf{n} into a bitstream of *at most* $(N-1)(H(\mathbf{p})+1)$ bits [9, Chap. 3], where $H(\cdot)$ computes the Shannon entropy of the probability mass function (pmf), and $\mathbf{p} = (p_{-K}, \dots, p_K)$ is the pmf of \mathbf{n} , i.e., $p_k = \#\{i \in \{1, \dots, N-1\} \mid n_i = k\} / (N-1)$. Note that if we directly encode \mathbf{n} without using any entropy coder, then $(N-1) \log(2K+1)$ bits is required, which is the worst case scenario. To encode $\hat{\sigma}^2$, which must be computed by (28) with the quantized DST coefficients, we again need to approximate it with precision δ , which costs $L^*([\hat{\sigma}^2/\delta])$ bits. Therefore, we have

$$L(\hat{\theta}_{\tilde{\beta}}, \hat{\sigma}^2 \mid \delta) \approx L^*(K) + (N-1)(H(\mathbf{p})+1) + L^*([\hat{\sigma}^2/\delta]). \quad (36)$$

Therefore, using (29), (34), and (36), together with $L(\delta) = \log(1/\delta)$, the total codelength $QMDL$ of (33) can be written as

$$\begin{aligned} QMDL(\delta) &= \log(1/\delta) + L^*([\alpha_0/\delta]) + L^*([\alpha_1/\delta]) \\ &\quad + L^*(K) + (N-1)(H(\mathbf{p})+1) \\ &\quad + L^*([\hat{\sigma}^2/\delta]) + \frac{N-1}{2} \log(2\pi e \cdot \hat{\sigma}^2), \end{aligned} \quad (37)$$

where \mathbf{p} , $\hat{\sigma}^2$, and K all depend on δ . We then search $\delta = \delta^*$ that minimizes (37) over a finite set of possible values. 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.

6. ADAPTIVE HIERARCHICAL SEGMENTATION, COMPRESSION, AND DENOISING

Based on our analysis of the global 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 now assume that $N = 2^J$ for some $J \in \mathbb{N}$, and let us define a collection of the standard dyadic subintervals on $[0, 1]$, $\mathcal{I}_J \triangleq \{I_{j,k} = [k/2^j, (k+1)/2^j] \mid j = 0, 1, \dots, J-1, k = 0, 1, \dots, 2^j - 1\}$. Let $N_j \triangleq 2^{J-j}$. The number of available samples on $I_{j,k}$ including the two endpoints is $N_j + 1$, which is common for all $k = 0, \dots, 2^j - 1$ for a given level j . Thus each of the shortest subintervals $I_{J-1,k}$ contains three samples.

We adopt the “split-and-merge” or “divide-and-conquer” approach à la the best basis algorithm 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 formula (32) (or (33) for the quantized approach) for $I_{j,k}$ instead

of the whole interval I . 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. However, 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 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 added to the cost of the union.
- Additional 2 bits must be added to the cost of the union that should be used as an indicator for the use of the union instead of the parent.
- For the quantization version, the precision δ should be level dependent, i.e., $\delta = \delta_j$. It is clear that the shorter the subinterval is, the lower the precision should be. Thus, we must *recompute* QMDL of the union of the two children with the same precision as the parent to be a fair comparison.

7. NUMERICAL RESULTS

We now demonstrate our algorithms discussed in the previous sections. Figure 1 shows the result of our “analytical” MDL segmentation, compression, and denoising algorithm. The original signal consists of piecewise smooth components with several jumps in the signal values and the derivatives, and a textured region toward the end. It also contains weak white Gaussian noise (i.e., $\sigma^2 = 10^{-14}$, which is of course unknown to our algorithms). Our result in Figure 1 features: almost perfect reconstruction of the smooth regions; progressively narrower subdivisions around the singularities (i.e., similar to the so-called Whitney decomposition); and visually-pleasing least squares fitting with trigonometric polynomials with linear functions over the textured regions. In this figure, our algorithm segmented this input signal into 25 segments (i.e., 26 endpoints), and retained 55 DST coefficients of the residual components in total, which resulted in 81 floating-point values. Since the original data consists of 257 floating-point values, the compression rate is not really high in this case, i.e., just about 3.17. The relative ℓ^2 error is 0.0922. Figure 2 shows the result of our quantized MDL algorithm. It generated a compressed representation of only 110 bytes whereas the original data size is 2056 bytes ($= 257 \times 8$ bytes), i.e., the compression ratio is about 19, which is quite significant (1/19th of the original file size). Yet the reconstruction is quite faithful to the original. The relative ℓ^2 error is 0.0114. Thus in terms of the quantitative performance, the quantized approach is superior to the analytical one, as we expected. One drawback of the quantized approach, however, is the difficulty of the search of the optimal δ_j^* . We used a very simple search around $\delta = 1/\sqrt{N+1}$ in our experiments, which resulted in less perfect split around the block discontinuities in the earlier part of the signal compared to the analytical approach. We are currently investigating a better search strategy for δ_j^* .

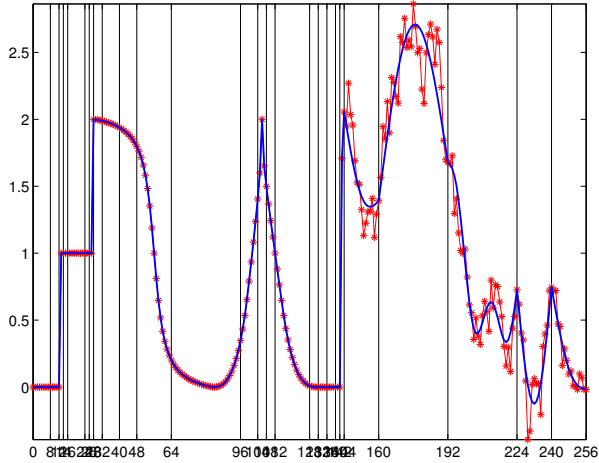


Fig. 1. The reconstructed signal using our analytical PHLST-MDL algorithm. The partitions obtained by our algorithm are shown as the vertical lines. The original data points are shown as connected *, and the recovered signal is overlaid as a blue curve.

The superiority of both of our approaches to the more conventional method is apparent if we compare the above figures with Figure 3, which is obtained by the best sparsifying basis (BSB) [10] using the $\ell^{0.1}$ sparsity norm and the local cosine dictionary. The partition pattern obtained by LCT-BSB is not too intuitive and the reconstructed signal from its top 81 largest coefficients shown in the figure reveals spurious Gibbs oscillations around the singularities. The relative ℓ^2 error is 0.0459. We are currently conducting our comparative study with the other MDL approaches using the conventional wavelets (e.g., [6, 8]), which we shall report at another opportunity.

8. 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., 2003, vol. Proc. SPIE 5207, pp. 495–506.
- [2] N. Saito and J.-F. Remy, "The polyharmonic local sine transform: A new tool for local image analysis and synthesis without edge effect," *Applied and Computational Harmonic Analysis*, 2005, To appear.
- [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] C. Lanczos, *Discourse on Fourier Series*, Hafner Publishing Co., New York, 1966.
- [5] J. Rissanen, *Stochastic Complexity in Statistical Inquiry*, World Scientific, Singapore, 1989.
- [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] N. Saito, "Simultaneous noise suppression and signal compression using a library of orthonormal bases and the min-

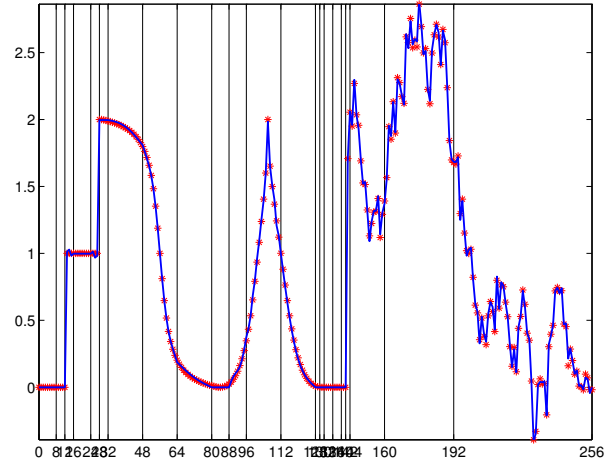


Fig. 2. The reconstructed signal using our quantized MDL algorithm (blue curve) vs the original data (red *s).

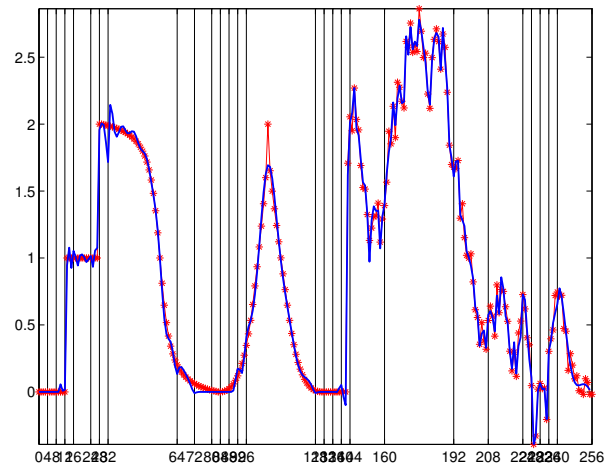


Fig. 3. The partition pattern computed by LCT-BSB algorithm as well as the reconstruction from its top 81 coefficients.

- imum description length criterion," in *Wavelets in Geophysics*, E. Foufoula-Georgiou and P. Kumar, Eds., chapter XI, pp. 299–324. Academic Press, San Diego, CA, 1994.
- [8] 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.
- [9] K. Sayood, *Introduction to Data Compression*, Morgan Kaufmann Publishers, Inc., San Francisco, CA, 2nd edition, 2000.
- [10] B. Bénichou and N. Saito, "Sparsity vs. statistical independence in adaptive signal representations: A case study of the spike process," in *Beyond Wavelets*, G. V. Welland, Ed., vol. 10 of *Studies in Computational Mathematics*, chapter 9, pp. 225–257. Academic Press, San Diego, CA, 2003.