Edge Preserving Image Regularization Based on Morphological Wavelets and Dyadic Trees

Zhen James Xiang, *Student Member, IEEE,* and Peter J. Ramadge, *Fellow, IEEE*

***Abstract*—Despite the tremendous success of wavelet based image regularization, we still lack a comprehensive understanding of the exact factor that controls edge preservation and a principled method to determine the wavelet decomposition structure for dimensions greater than 1. We address these issues from a machine learning perspective by using tree classifiers to underpin a new image regularizer that measures the complexity of an image based on the complexity of the dyadic tree representations of its sublevel sets. By penalizing unbalanced dyadic trees less, the regularizer preserves sharp edges. The main contribution of the paper is the connection of concepts from structured dyadic tree complexity measures, wavelet shrinkage, morphological wavelets, and smoothness regularization in Besov space into a single coherent image regularization framework. Using the new regularizer, we also provide a theoretical basis for the data-driven selection of an optimal dyadic wavelet decomposition structure. As a specific application example, we give a practical regularized image denoising algorithm that uses this regularizer and the optimal dyadic wavelet decomposition structure.**

***Index Terms*—Wavelet transforms, morphological operations, image enhancement, multidimensional signal processing.**

I. Introduction and Prior Work

In a typical image regularization problem, one seeks to minimize the sum of two terms. The first is an error metric measuring the error between the signal estimate f and a noisy observation $\tilde{f}$; the second is a regularization term measuring the complexity of the estimate f:

$$\text{minimize } \|f - \tilde{f}\|_U^2 + \lambda \|f\|_V. \tag{1}$$

An important challenge is selecting these measures to preserve sharp edges and other meaningful high frequency features in images while also ensuring that (1) can be efficiently solved. This requires the regularization measure $\|\cdot\|_V$ to tolerate sharp edges, so that an image f with sharp edges can yield a low $\|f\|_V$ value and minimize (1). Traditional regularization methods (e.g. $\|\cdot\|_{L_2}$) fail to meet this criterion because the selected $\|\cdot\|_V$ is agnostic to the pixel locations and the edge structure of f [1].

The conventional solution to the above challenge is through wavelet-based methods [2]–[4], in which $\|\cdot\|_V$ is the l_1 norm of the wavelet coefficients. This approximates the Besov norm in Besov space [5]–[8]. Despite the huge success and popularity of such methods, we still lack (a) a comprehensive understanding of the exact factor that controls edge preservation and (b) a principled method to determine the wavelet decomposition structure for dimensions greater than 1.

We address these issues from a machine learning perspective by using tree classifiers [9]–[11] to underpin a new image regularizer. In our framework, $\|\cdot\|_V$ measures the complexity of an image based on the complexity of the dyadic tree representations [12] of its sublevel sets. The unbalanced structure of these dyadic trees is key to edge preservation. The main contribution of the paper is the connection of concepts from structured dyadic tree complexity measures [12]–[15], wavelet shrinkage [3], [4], morphological wavelets [16], [17] and smoothness regularization in Besov space [6]–[8] into a single coherent image regularization framework, in which the degree of edge preservation is controlled by a single parameter s. The theoretical connections provide intuitive ways to understand this parameter. The framework also exploits the structural flexibility of trees to define and adaptively select an optimal "wavelet" decomposition structure in 2D spaces. Hence the regularization framework supports data adaptive results.

Our approach is inspired by recent developments in dyadic decision tree regularization [12]–[15] and its application to the estimation of sublevel sets with sharp boundaries [18], [19]. We extend the regularizer in [18], [19], which only applies to sublevel sets (i.e. binary valued functions), to a regularizer that applies to any real valued 2D or higher dimensional signal. To our best knowledge, this is the first work to use the dyadic decision tree regularization in a general real valued signal regularization setting. Our approach focuses on a mathematical and algorithmic analysis of the proposed framework. This paper is an integration and significant extension of preliminary investigations [20], [21]. Specifically, the connection with Besov space is entirely new, all proofs are more general, and experimental results are more extensive.

The paper is organized as follows. In Section II, we briefly review the background of the various approaches that we will bring together. Using 1D signals, we introduce the new regularizer in Section III, show its theoretical properties and connections (Section IV) and introduce effective algorithms to solve the regularized denoising problem (Section V). With these 1D preparations out of the way, we extend this regularization method to higher dimensional spaces in Section VI and apply it to image denoising. We report experimental results in Section VII and conclude in Section VIII.

II. Background

Let's return to equation (1) and give a more detailed account. The regularization term $\|\cdot\|_V$ is often associated

Zhen James Xiang is with the Department of Electrical Engineering, Princeton University, Princeton, NJ 08544 USA (e-mail: zxiang@princeton.edu).

Peter J. Ramadge is with the Department of Electrical Engineering, Princeton University, Princeton, NJ 08544 USA (e-mail: ramadge@princeton.edu).

with encouraging "smoothness" of f in some space V. The particular choice of $\|\cdot\|_V$ thus reflects our prior assumptions on the regularity of f. Several successful regularization methods have been extensively studied and are briefly reviewed below.

A. Wavelet and Morphological Wavelet Transforms

Wavelet transforms [2] have been tremendously successful in image denoising applications. For linear orthogonal wavelets, solving (1) with $\|\cdot\|_V$ being the l_0 (resp. l_1) norm of the wavelet coefficients simply requires hard (resp. soft) thresholding these coefficients (i.e., wavelet shrinkage methods [3], [4]). Most of the wavelet transforms used are linear, but there has been recent interest in non-linear extensions [22]. The morphological wavelet transform [16], [17] is a non-linear wavelet transform that replaces the algebraic operations in the Haar wavelet transform with $\max$ and $\min$ operators. This modification improves the preservation of edges in low resolution signals. Moreover, like most wavelets, it can be effectively computed in-place in a dyadic decomposition fashion (i.e., it is based on a dyadic tree). In our subsequent theoretical analysis (Section IV-B) we show that a new regularizer $\|\cdot\|_V$, constructed using tools developed for machine learning, has a natural and tight theoretical connection to the morphological Haar wavelet. Our use of the morphological wavelet is thus a tight consequence of linking a natural regularizer in machine learning to signal denoising.

B. Smoothness: Total Variation and Besov Space

It has also been of interest to determine what is the best smoothness space V (and norm $\|\cdot\|_V$) for natural images. One possible choice is bounded variation space and the total variation norm [1], which measures the total change in a signal: $\|f\|_{TV} = \|\bigtriangledown f\|_{L_1}$. This allows occasional jumps in the signal (sharp edges) but penalizes frequent oscillations.

An alternative smoothness space is Besov space. For any 1D function f with a finite support Ω, we first define the modulus of continuity with parameter $t > 0$ as:

$$w_n(f,t)_{\mathcal{L}^p} = \sup_{|h|\leq t} (\|\Delta_h^n f\|_{\mathcal{L}^p(\Omega_{h,n})}), \tag{2}$$

in which Δ_h^n is a n-th order difference operator defined by $\Delta_h f(x) = f(x) - f(x-h)$ and $\Delta_h^n f(x) = \Delta_h^{n-1} f(x) - \Delta_h^{n-1} f(x-h)$. $\Omega_{h,n} = \{x \in \Omega, x-kh \in \Omega, k = 0,1,\ldots,n\}$. The Besov space $B_{p,q}^s$ is then defined as

$$B_{p,q}^s = \left\{ f : \ f \in \mathcal{L}_p \text{ and } (2^{sj} w_n(f,2^{-j})_{\mathcal{L}^p})_{j\geq 0} \in l_q \right\}. \tag{3}$$

Here n is an integer larger than s. This space is associated with the following Besov seminorm:

$$|f|_{B_{p,q}^s} = \left\| (2^{sj} w_n(f,2^{-j})_{\mathcal{L}^p})_{j\geq 0} \right\|_{l_q}, \tag{4}$$

and the following Besov norm:

$$\|f\|_{B_{p,q}^s} = \|f\|_{\mathcal{L}^p} + |f|_{B_{p,q}^s}.$$

For more detailed discussion on Besov space, we refer readers to the introductions in [5], [6], [8], and the books of Triebel (e.g. [23]). Here we simply point out that in Besov space $B_{p,q}^s$, the parameters p, q, s all indicate levels of smoothness and such smoothness is a good mathematical representation for the "smoothness" in natural images. In particular, when $p < 2$, it tolerates the discrete singularities in signals corresponding to the sharp edges [2]. Besov space is the underlying smoothness space for the wavelet shrinkage method [3], [4], which approximately solves (1) with V the Besov space $B_{1,1}^1$ and with $\|\cdot\|_V$ the corresponding Besov norm [7]. This connection partly explains the success of the wavelet shrinkage method. Besov space has also been used in the learning community to study wavelet kernel regression [24] (with $p = 2$) and penalized empirical risk minimization [25] (with $p < 2$).

C. Regularized Dyadic Decision Trees

To illustrate the relevance of decision trees, [9]–[11], to image regularization, consider a tree classifier T that divides a decision space X into disjoint regions $X = S \cup \bar{S}$ (binary classification). Each node of T is a subset of X and leaf nodes are labeled 1 (in S) or 0 (in $\bar{S}$). The set S is the union of the leaf nodes labeled 1. The boundary of S can be regularized by penalizing a complexity measure $\Phi(T)$ of T. A more (resp. less) complex tree can yield a boundary that is more (resp. less) finely structured. Tree complexity is thus related to the edge structure of its decision region.

In machine learning, penalizing the complexity of a tree classifier is an important means of controlling overfitting [26], [27]. Various complexity measures have been proposed for this purpose since Breiman's seminal "Classification and Regression Trees" (CART) [10], in which tree complexity is measured by the number of leaf nodes. Scott and Nowak proved that dyadic decision trees are asymptotically optimal in the minimax sense when solved using such regularization [13]. However, this complexity measure is agnostic to tree shape. Trees with the same number of leaf nodes can still have notably different complexities [28], [29]. Recently, Scott and Nowak proposed a "spatially adaptive" complexity measure that prefers unbalanced trees and enables a more detailed local fit to a decision boundary. Theoretically, this complexity measure has better convergence properties [14], [15], and has been successfully applied to the estimation of sublevel sets [18], [19]. In effect, a complexity measure that favors unbalanced trees allows a finer representation of the edge structure of the decision region S.

III. A New Image Regularizer

We now introduce our new image regularizer. We first define a general parameterized complexity measure for dyadic decision trees; then use this to define the new complexity measure (i.e., regularizer) for real valued functions.

A. Measuring the Complexity of a Dyadic Decision Tree

Let T be a dyadic decision tree on the interval $[0,1)$. By this we mean that: (a) the root node of T is $[0,1)$ and (b) every non-leaf node $[x_1,x_2)$ of T has two children obtained by cutting the interval into halves, $[x_1,(x_1+x_2)/2)$ and $[(x_1+x_2)/2,x_2)$.

Let $\pi(T)$ denote the set of leaf nodes of T and consider the complexity measure

$$\Phi(T) = \sum_{L \in \pi(T)} \phi(|L|). \tag{5}$$

$\Phi(T)$ penalizes each leaf node $L \in \pi(T)$ according to its size $|L|$. In our setting, if $L \in \pi(T)$ is at level k, then $|L| = 2^{-k}$, $k \geq 0$. So $\Phi(T)$ is parameterized by the sequence $\{\alpha_k\}_{k=0}^{\infty}$ with $\alpha_k = \phi(2^{-k})$. We assume that $\alpha_0 = 0$, i.e., a single node tree has complexity 0.

This parameterized definition covers many previously proposed complexity measures as special cases. For example, setting $\alpha_k = 1$, $k > 0$, yields the complexity measure of CART [10]: $\Phi(T) = |\pi(T)|$. Under this setting, trees (b) and (c) in Figure 1 have the same complexity. On the other hand, setting $\alpha_k = \sqrt{2^{-k}(C_1 + C_2 k)}$ yields the complexity measure of [18], [19] that favors unbalanced trees, and setting $\alpha_k = k2^{-k}$ yields the example-weighted average tree depth measure proposed in [28]. In these cases, tree (c) is measured as less complex than tree (b). However, instead of fixating on a specific sequence, we simply require $\{\alpha_k\}_{k=0}^{\infty}$ to satisfy the following two conditions.

Condition A (Preference for Simplicity):

$$\forall k \geq 1, \alpha_k > \frac{1}{2}\alpha_{k-1}.$$

This ensures that splitting a node *increases* complexity: $\phi(2^{-k}) + \phi(2^{-k}) > \phi(2^{-(k-1)})$. So $\Phi(T)$ favors simpler trees.

Condition B (Preference for Unbalanced Trees):

$$\exists c > 0, \varepsilon > 0 \text{ s.t. } \forall k \geq 1 : \alpha_k < c\varepsilon^k.$$

This requires $\{\alpha_k\}_{k=1}^{\infty}$ to decay geometrically. So deeper nodes in the tree receive less weight. This gives a tree the flexibility to make a detailed fit around a sparse set of edges which in turn gives rise to an unbalanced tree structure. This connection is explored further in Appendix A.

We often use the following re-parameterization:

$$\begin{aligned} \alpha_0 &= 0, & (6)\\ \alpha_k &= 2^{-(1-s)k}, k \geq 1. & (7) \end{aligned}$$

The scalar $s \in [0, 1]$ is the most important parameter in the paper. It encodes the characteristics of the complexity measure and controls the preservation of edges and high frequency features. Conditions A and B dictate that $s > 0$ and $s < 1$, respectively. $s = 1$ yields the complexity measure $\Phi(T) = |\pi(T)|$ of CART [10]. Reducing s decreases the cost of deeper nodes and this in turn reduces the cost of unbalanced dyadic trees.

B. Measuring the Complexity of a Real valued Function

From the simple illustration in Section II-C it is clear how to use $\Phi(T)$ to measure the complexity of a binary valued function. We now extend this to a complexity measure of bounded real valued functions. The basic idea is to measure the complexity of a real valued function by integrating the complexity of the indicator functions of its sublevel sets.

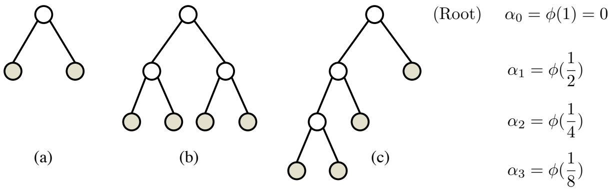

(Root)
$\alpha_0 = \phi(1) = 0$
$\alpha_1 = \phi(\frac{1}{2})$
$\alpha_2 = \phi(\frac{1}{4})$
$\alpha_3 = \phi(\frac{1}{8})$
(a)
(b)
(c)

Fig. 1. Illustration of the complexity measure (5). Leaf nodes (gray circles) at level k are weighted by $\alpha_k = \phi(2^{-k})$. The complexity measures for tree (a)-(c) are $2\alpha_1$, $4\alpha_2$, and $\alpha_1 + \alpha_2 + 2\alpha_3$, respectively. Tress (b) and (c) have the same number of leaf nodes but can have different complexity measures.

To formally describe this process, let $f : [0, 1) \mapsto \mathbb{R}$ be a given bounded signal. Fix an integer $m > 1$. For each threshold γ, we represent the sublevel set $S_\gamma = \{x \in [0, 1) : f(x) \leq \gamma\}$ using a dyadic tree of maximum depth m as follows. The root node (height 0) of the tree is $[0, 1)$. If a node, $[x_1, x_2)$, of height i ($i < m$) is not entirely inside or outside S_γ, (i.e. $[x_1, x_2) \backslash S_\gamma \neq \phi, S_\gamma \cap [x_1, x_2) \neq \phi$), we split the node into halves ($[x_1, (x_1 + x_2)/2)$ and $[(x_1 + x_2)/2, x_2)$). We repeat this process until either every node is entirely inside or outside S_γ or is at depth m. Nodes at depth m are not split. Denote the resultant finite tree as T_γ^m. The complexity of S_γ is measured by the complexity of its dyadic tree representation $\Phi(T_\gamma^m)$ defined in (5). Now define the complexity of f as an integral over γ of the complexities of its sublevel sets:

$$E_m(f) = \int \Phi(T_\gamma^m) d\gamma. \tag{8}$$

By the assumption $\alpha_0 = 0$, $\Phi(T_\gamma^m) = 0$ if γ is outside the range of f. Therefore the integral is always finite because $\Phi(T_\gamma^m)$ is finite and f is bounded.

IV. Properties and Connections

We now study the theoretical properties of the complexity measure (8) and its connections to dynamic range, wavelet-based methods, and Besov space. In particular, we show that the parameter s controls edge preservation and discuss how it manifests itself in these different domains.

A. Connection to Dynamic Range

$E_m(f)$ can be written as a weighted summation of the dynamic ranges of f on a series of intervals. For $0 \leq k < m, 0 \leq l < 2^k$, define the following intervals:

$$I_{k,l} = \{x : 0 \leq 2^k x - l < 1\}. \tag{9}$$

The dynamic range of f on $I_{k,l}$ is:

$$d_{k,l}(f) = \sup_{x \in I_{k,l}} f(x) - \inf_{x \in I_{k,l}} f(x). \tag{10}$$

We then have the following result.

Theorem 1.

$$E_m(f) = \sum_{k=0}^{m-1} \sum_{l=0}^{2^k - 1} (2\alpha_{k+1} - \alpha_k) d_{k,l}(f). \tag{11}$$

The general proof is included in Appendix B. A brief proof for continuous functions f is in Section 4 of [20].

In the right hand side of (11), the inner summation $\sum_{l=0}^{2^k-1} d_{k,l}(f)$ sums the dynamic ranges of f on the 2^k small intervals $I_{k,l},\ 0 \le l < 2^k$. This summation is a coarse measure of the oscillation of f at resolution 2^k. By Theorem 1, $E_m(f)$ measures this oscillation in f across multiple resolutions.

Corollary 1. *$E_m(f)$ is a convex function over bounded $f\colon [0,1) \to \mathbb{R}$ if and only if $2\alpha_{k+1} - \alpha_k > 0,\ 1 \le k < m$.*

Corollary 1 is proved in Appendix C. The corollary connects the convexity of $E_m(f)$ to Condition A, the preference for simplicity. When using $E_m(f)$ to regularize a convex objective, the regularized optimization problem will be convex if and only if $\Phi(T)$ prefers simpler trees - a natural minimal requirement for a complexity measure.

B. Connection to Morphological Wavelet Coefficients

In this subsection we assume that the signal of interest has maximum resolution $N = 2^m$, i.e., it is constant on the intervals $I_{m,l}$, defined in (9), for $0 \le l \le N-1$. To emphasize this, we use f_m, instead of f, to denote the signal.

Any f_m can be mapped to a vector in $\Psi_m \in \mathbb{R}^N$ by setting $\Psi_m(l) = f_m(x)|_{x \in I_{m,l}}$. We can then compute a wavelet decomposition of Ψ_m. For example, the Haar wavelet decomposition is computed as:

$$\Psi_k(l) = \frac{\Psi_{k+1}(2l) + \Psi_{k+1}(2l+1)}{2}, 0 \le l \le 2^k - 1, \quad (12)$$

$$W_{k,l} = \Psi_{k+1}(2l) - \Psi_{k+1}(2l+1), 0 \le l \le 2^k - 1. \quad (13)$$

In the above formulae, k takes decreasing values from $m-1$ to 0, and indexes a chain of approximation signals $\Psi_k \in \mathbb{R}^{2^k}$ with decreasing resolution (and hence decreasing length). $W_{k,l}$ are the wavelet coefficients.

We will now introduce a pivotal theorem that connects $E_m(f_m)$ with a specific kind of wavelet transform called the morphological Haar wavelet transform. This nonlinear transform replaces the averaging operator in (12) by either a $\max$ (denoted $\vee$) or $\min$ (denoted $\wedge$) operator. The $\max$ version of the transform is:

$$\Psi_k^{\vee}(l) = \Psi_{k+1}^{\vee}(2l) \vee \Psi_{k+1}^{\vee}(2l+1), 0 \le l \le 2^k - 1, \quad (14)$$

$$W_{k,l}^{\vee} = \Psi_{k+1}^{\vee}(2l) - \Psi_{k+1}^{\vee}(2l+1), 0 \le l \le 2^k - 1, \quad (15)$$

and the $\min$ version is:

$$\Psi_k^{\wedge}(l) = \Psi_{k+1}^{\wedge}(2l) \wedge \Psi_{k+1}^{\wedge}(2l+1), 0 \le l \le 2^k - 1, \quad (16)$$

$$W_{k,l}^{\wedge} = \Psi_{k+1}^{\wedge}(2l) - \Psi_{k+1}^{\wedge}(2l+1), 0 \le l \le 2^k - 1. \quad (17)$$

We have the following important equivalency result:

Theorem 2. *Let f_m be a 1D signal of maximum resolution 2^m and $\{W_{k,l}^{\vee}\}$, $\{W_{k,l}^{\wedge}\}$ be defined by* (14)-(17). *Then*

$$E_m(f_m) = \sum_{k=0}^{m-1} \sum_{l=0}^{2^k-1} \alpha_{k+1} \left(|W_{k,l}^{\vee}| + |W_{k,l}^{\wedge}| \right). \quad (18)$$

A brief proof of Theorem 2 can be found in Section 4 of [20]. A complete, detailed proof is included in Appendix D.

Regularization using $E_m(f_m)$ controls the complexity of the sublevel sets of f_m by penalizing $\Phi(T_\gamma^m)$. By Theorem 2, this is equivalent to a weighted l_1 norm of the morphological wavelet coefficients of f_m. Thus our new regularization method is fundamentally connected to the l_1 wavelet shrinkage method proposed by Donoho and Johnstone [3], [4], and the sparse representation based image denoising methods in general. In these methods sparsity is not limited to any specific wavelet type. However, from Theorem 2 we see that structured dyadic tree regularization used in machine learning yields a connection to the sparsity of the coefficients of one specific wavelet: the Haar morphological wavelet. Theorem 2 also offers a key insight on how level-adpative wavelet thresholding impacts edge preservation. Consider the parameterization (6), (7). At $s = 1$, all wavelet coefficients are weighted equally: $\alpha_k = 1,\ 1 < k < m$. This corresponds to traditional wavelet shrinkage (same threshold for every coefficient), and the traditional tree complexity measure $\Phi(T) = |\pi(T)|$. As s decreases ($0 < s < 1$), $\Phi(T)$ allows trees to adapt their resolution to the edges of the signal. By Theorem 2, this corresponds to wavelet-thresholding with decreasing thresholds on higher resolution coefficients and suggests that this level-adaptive shrinkage will improve the resultant resolution around edges. However, when α_k decreases so fast that $s = 0$ (i.e. $\alpha_k = \frac{\alpha_{k-1}}{2}$), the scheme fails since $\Phi(T)$ no longer prefers simpler trees. Empirical examples for various values of s are given in [20].

C. Connection to Besov Space

In computing $E_m(f)$ we are approximating the sublevel sets of f with trees of maximum depth m. As $m \to \infty$, the trees grow in order to approximate the sublevel sets of f as accurately as possible. Using the parameterization (6), (7), we now show that if the resultant sequence $\{E_m(f)\}$ converges, then it converges to an upper bound of the Besov seminorm determined by s.

Theorem 3. *Let $f : [0,1) \to \mathbb{R}$ be any bounded function. If the sequence $\{E_m(f)\}_{m=1}^{\infty}$ converges, then f is in the Besov space $B_{1,1}^s$ defined in* (3), *and*

$$C_1 |f|_{B_{1,1}^s} \le \lim_{m \to \infty} E_m(f) \quad (19)$$

where $|f|_{B_{1,1}^s}$ is defined in (4) *and C_1 is a constant.*

This theorem is proved in Appendix E.

Besov space and Besov norms have emerged as useful mathematical tools for modeling the smoothness of natural images [6]–[8]. In this light, Theorem 3 shows a connection between $E_m(f)$ and Besov space and gives assurance of the proposed new regularizer's performance on a important class of image denoising tasks.

Finally, we provide some consistency results on finite sampling. For a signal f defined on a continuous interval $[0,1)$, we compute $E_m(\cdot)$ on a "sampled" version of f. The following theorem shows that as the sampling resolution goes to infinity, the result of a sampled computation converges to $E_\infty(f)$. This theorem is proved in Appendix F.

Theorem 4. *Let $f : [0,1) \to \mathbb{R}$ be a bounded function in $B_{1,1}^s$ ($0 \le s \le 1$) and assume that $\{E_m(f)\}_{m=1}^{\infty}$ converges.*

For a given m, let $\mathcal{X}_m = \{x_l \mid 0 \le l < 2^m, x_l \in I_{m,l}\}$ be a set of sample points and $f_{\mathcal{X}_m}(x)$ be the sampled signal: $f_{\mathcal{X}_m}(x) = f(x_l)$ for $x \in I_{m,l}$. We have

$$\lim_{m\to\infty} \sup_{\mathcal{X}_m} E_m(f_{\mathcal{X}_m}) = \lim_{m\to\infty} E_m(f). \tag{20}$$

Moreover, if f is also σ-Hölder continuous for some $\sigma > s$ (i.e. $\exists C$ s.t. $|f(x) - f(y)| \le C|x-y|^\sigma, \forall x, y \in [0,1)$), then

$$\lim_{m\to\infty} E_m(f_{\mathcal{X}_m}) = \lim_{m\to\infty} E_m(f). \tag{21}$$

This is stronger than (20) *because now $E_m(f_{\mathcal{X}_m})$ converges to $E_\infty(f)$ regardless of the choice of sample points.*

D. Edge Preservation and Parameter s

The connections established in this section can be summarized as follows. The continuous parameter $s \in [0,1]$ controls edge preservation. For $s = 1$: $\alpha_k = 1$ $(k \ge 1)$; nonroot leaf nodes of the tree regularizer are penalized equally; and morphological wavelet coefficients are penalized equally across scales (18). As s is decreased: $\alpha_k = 2^{-(1-s)k}$ decays exponentially in $k \ge 1$; deeper leaf nodes of the tree regularizer are penalized less; and less penalty is applied to finer scale morphological coefficients (18). A smaller s this allows better representation of edges. Correspondingly, s fine tunes the smoothness space $B^s_{1,1}$. For $s_1 < s_2$, $B^{s_2}_{1,1} \subset B^{s_1}_{1,1}$. So as s decreases, $E_\infty(\cdot)$ connects to the Besov seminorm in a larger Besov space $B^s_{1,1}$ and the convergence of $\{E_m(f)\}^\infty_{m=1}$ implies that f is contained in this larger Besov space. This indicates a less stringent smoothness requirement.

V. Regularized Denoising Problems

We now solve the regularized denoising problem (1) using the complexity measure $E_m(\cdot)$ as a regularizer. Since this involves computation, throughout the section we assume that each signal f_m has maximum resolution $N = 2^m$, and is represented as a vector in $\mathbb{R}^N$: $\mathbf{f}_m(l) = f(x)|_{x\in I_{m,l}}$. Given a noisy observation $\tilde{\mathbf{f}}_m$, we consider the denoising problem:

$$\min_{\hat{\mathbf{f}}_m \in \mathbb{R}^N} \quad \|\hat{\mathbf{f}}_m - \tilde{\mathbf{f}}_m\|_2^2 + \lambda E_m(\hat{f}_m). \tag{22}$$

This problem can be reformulated as a Second Order Cone Programming (SOCP) problem. To see this, we first introduce some dummy variables. Let $\varepsilon = \hat{\mathbf{f}}_m - \tilde{\mathbf{f}}_m$ have l-th component ε_l, $l = 0, \ldots, N-1$, and $e = \sum_{l=0}^{N-1} \varepsilon_l^2 = \|\hat{\mathbf{f}}_m - \tilde{\mathbf{f}}_m\|_2^2$. We relax this equality to the following inequality

$$e \ge \sum_{l=0}^{N-1} \varepsilon_l^2 \iff (\frac{e+1}{2}, \frac{e-1}{2}, \varepsilon_0, \ldots, \varepsilon_{N-1}) \ge_{\mathcal{K}} 0, \tag{23}$$

where $\ge_{\mathcal{K}}$ denotes a second order Lorentz cone inequality. Equation (23) is a standard form SOCP constraint. Now consider the perfect binary tree $\mathfrak{T}$ with: 1) Root node $I_{0,0} = [0,1)$; 2) Each non leaf node $I_{k,l}$ has children $I_{k+1,2l}$ and $I_{k+1,2l+1}$, corresponding to the left/right halves of $I_{k,l}$ respectively; 3) Leaf nodes are the intervals $I_{m,l}$, $l = 0, \ldots, N-1$. For every edge $b \in \mathfrak{T}$ connecting parent I_p with child I_c, define:

$$\begin{aligned} \delta_b^\vee &= \quad (\max_{x\in I_p} \hat{f}(x) - \max_{x\in I_c} \hat{f}(x)), \\ \delta_b^\wedge &= -(\min_{x\in I_p} \hat{f}(x) - \min_{x\in I_c} \hat{f}(x)). \end{aligned}$$

These variables are nonnegative:

$$\forall b \in \mathfrak{T}: \qquad \delta_b^\vee \ge 0, \quad \delta_b^\wedge \ge 0, \tag{24}$$

and for any path $\mathcal{P}$ in $\mathfrak{T}$ from $I_{k,l}$ to a leaf node L:

$$\varepsilon_L + \tilde{f}_m|_L + \sum_{b\in\mathcal{P}} \delta_b^\vee \;=\; \max_{x\in I_{k,l}} \hat{f}(x) \tag{25}$$

$$\varepsilon_L + \tilde{f}_m|_L - \sum_{b\in\mathcal{P}} \delta_b^\wedge \;=\; \min_{x\in I_{k,l}} \hat{f}(x) \tag{26}$$

These dependancies can be captured by considering just "left" and "right" paths in $\mathfrak{T}$. The left path $\mathcal{P}^U_{k,l}$ from non-leaf node $I_{k,l}$ always proceeds to left children, ending at leaf node $\mathcal{L}^U_{k,l} = I_{m,\theta^U_{k,l}}$. Similarly, its right path $\mathcal{P}^V_{k,l}$, always proceeds to right children, ending at leaf node $\mathcal{L}^V_{k,l} = I_{m,\theta^V_{k,l}}$. By (25), (26), for $0 \le k \le m-1$ and $0 \le l \le 2^k - 1$:

$$\varepsilon_{\theta^U_{k,l}} + \tilde{f}_m|_{\mathcal{L}^U_{k,l}} + \sum_{b\in\mathcal{P}^U_{k,l}} \delta_b^\vee = \varepsilon_{\theta^V_{k,l}} + \tilde{f}_m|_{\mathcal{L}^V_{k,l}} + \sum_{b\in\mathcal{P}^V_{k,l}} \delta_b^\vee, \tag{27}$$

$$\varepsilon_{\theta^U_{k,l}} + \tilde{f}_m|_{\mathcal{L}^U_{k,l}} - \sum_{b\in\mathcal{P}^U_{k,l}} \delta_b^\wedge = \varepsilon_{\theta^V_{k,l}} + \tilde{f}_m|_{\mathcal{L}^V_{k,l}} - \sum_{b\in\mathcal{P}^V_{k,l}} \delta_b^\wedge. \tag{28}$$

We can now prove the following theorem.

Theorem 5. *The solution of the SOCP problem:*

$$\textit{minimize} \qquad e + \lambda \sum_{k=0}^{m-1} \sum_{\substack{\textit{edges } b\in\mathfrak{T} \textit{ that} \\ \textit{start at depth } k}} \alpha_{k+1} \left(\delta_b^\vee + \delta_b^\wedge\right), \tag{29}$$

$$\textit{subject to} \qquad (23),\ (24),\ (27) \textit{ and } (28).$$

yields the solution of the denoising problem (22).

Proof: Let J_{SOCP} (resp. J_{D}) denote the optimal value of (29) (resp. (22)). For a solution $\hat{\mathbf{f}}_m$ of (22), $\{\delta_b^\vee, \delta_b^\wedge, \varepsilon_l, e \colon b \in \mathfrak{T}, l = 0, \ldots, N-1\}$ satisfy (23)-(28). Hence $J_{\text{SOCP}} \le J_{\text{D}}$. Now let $\{\delta_b^\vee, \delta_b^\wedge, \varepsilon_l, e \colon b \in \mathfrak{T}, l = 0, \ldots, N-1\}$ be a solution of (29). Let $\hat{\mathbf{f}}_m = \tilde{\mathbf{f}}_m + \varepsilon$ and $\hat{W}^\vee_{k,l}$, $\hat{W}^\wedge_{k,l}$ denote the morphological wavelet coefficients of $\hat{\mathbf{f}}_m$. Optimality implies (23) is an equality, i.e., $e = \|\hat{\mathbf{f}} - \tilde{\mathbf{f}}_m\|_2^2$. Optimality and constraints (24), (27), (28), imply that (25), (26) hold. From this it follows that for any non-leaf node $I_{k,l}$ and edges b_U, b_V that connect to its children, one of $\delta^\vee_{b_U}$, $\delta^\vee_{b_V}$ must be 0. Hence $\delta^\vee_{b_U} + \delta^\vee_{b_V} = |\delta^\vee_{b_U} - \delta^\vee_{b_V}| = |\hat{W}^\vee_{k,l}|$. Similarly $\delta^\wedge_{b_U} + \delta^\wedge_{b_V} = |\hat{W}^\wedge_{k,l}|$. So $J_{\text{SOCP}} = \|\hat{\mathbf{f}} - \tilde{\mathbf{f}}_m\|_2^2 + \sum_{k=0}^{m-1} \sum_{l=0}^{2^k-1} \alpha_{k+1}(|\hat{W}^\vee_{k,l}| + |\hat{W}^\wedge_{k,l}|) = \|\hat{\mathbf{f}} - \tilde{\mathbf{f}}_m\|_2^2 + E_m(\hat{f}_m)$ (Theorem 2). Therefore $J_{\text{D}} \le J_{\text{SOCP}}$. ■

Problem (29) is a SOCP of size $\mathcal{O}(N)$, where N is the number of signal samples, and can be solved efficiently with existing SOCP toolboxes. Some examples are given in [20].

We will not dwell on the 1D case further. Instead, we now proceed to generalize these results to higher dimensions. This leads to the application of the proposed regularizer to the problem of image denoising.

VI. Extension to Higher Dimensional Spaces

The major challenge in extending the regularizer to higher dimensions is the non-uniqueness of the dyadic splitting. For example, consider representing sublevel set $S_\gamma = \{x \in [0,1)^d : f(x) \le \gamma\}$ in $\mathbb{R}^d$, $d > 1$, using a dyadic tree. We start with root node $[0,1)^d$ and if a node is not entirely inside or outside S_γ, we split the node. But now there are d distinct ways to do the splitting; which one should we choose?

A similar issue arises in computing a multidimensional Haar wavelet or morphological Haar wavelet transform. This recursively takes two signal points and computes the approximating signal Φ and wavelet coefficient W (see (12)-(17)). To order these computations for multidimensional signals, it is necessary to fix a dyadic structure for the computation, e.g. for images, we might do rows first then columns. However, fixing a dyadic structure a priori is usually an arbitrary decision.

We first establish our previous results on a fixed dyadic structure in d dimensions (Section VI-A). Then we pose and solve the problem of finding the optimal dyadic structure for a given signal (Section VI-B). Finally, we propose a denoising framework that jointly optimizes the dyadic structure and the signal estimation (Section VI-C). From a wavelet perspective, this provides a theoretical basis for the data-driven selection of a wavelet decomposition structure.

A. Generalization Under A Fixed Dyadic Structure

We first define a **full dyadic tree** and use it to represent a fixed dyadic structure. A binary tree $\mathfrak{T}$ is called a full dyadic tree on $[0,1)^d$ with resolution 2^m if it satisfies the following:

P1: The root node of $\mathfrak{T}$ is $J_{0,0} = [0,1)^d$.

P2: Any non-leaf node $J_{k,l}$ has a color $c \in \{1, 2, \dots, d\}$. Its children, $J_{k+1,2l}$ and $J_{k+1,2l+1}$, are congruent hyperrectangles obtained by cutting $J_{k,l}$ with the $d-1$ dimensional hyperplane through the center of $J_{k,l}$ and perpendicular to axis-c.

P3: All leaf nodes are hypercubes of side length 2^{-m}.

By **P3**, $\mathfrak{T}$ has height dm and by **P2**, each hyperrectangle at depth k has volume $|J_{k,l}| = 2^{-k}$ $(0 \le k \le dm)$. When $d = 1$, every node has only one color and $\mathfrak{T}$ reduces to the dyadic partition tree of the 1-D interval $[0,1)$ defined in Section V.

Assume that a signal of interest, denoted by f_m, is bounded and has maximum resolution 2^m in each dimension. So f_m is constant on the hypercubes $J_{dm,l}$, $l = 0, \dots, 2^{dm}-1$. For each threshold γ, represent the sublevel set $S_\gamma = \{x \in [0,1) : f_m(x) \le \gamma\}$ using a dyadic tree, rooted at $J_{0,0} = [0,1)^d$. Whenever a node J is not entirely inside or outside S_γ, we split J in exactly the same way that J is split in $\mathfrak{T}$. Denote the resulting tree as T_γ^m. Then following (8), for a fixed dyadic structure $\mathfrak{T}$, the complexity of f_m on $\mathfrak{T}$ is:

$$E_{m,\mathfrak{T}}(f_m) = \int \Phi(T_\gamma^m) d\gamma.$$

This is finite since the range of f_m and $\Phi(T_\gamma^m)$ are finite.

Analogous to (10), we define the dynamic range of f_m on hyperrectangle $J_{k,l}$ as

$$d_{k,l}(f_m) = \max_{x \in J_{k,l}} f_m(x) - \min_{x \in J_{k,l}} f_m(x).$$

Notice that $d_{k,l}(f_m)$ is implicitly dependent on $\mathfrak{T}$ although for notational simplicity we have omitted $\mathfrak{T}$ from the subscript.

Define the Haar morphological wavelet transform on the fixed dyadic structure $\mathfrak{T}$ as follows. The highest resolution signal Ψ_{dm} takes the values on leaf nodes $J_{dm,l}$: $\Psi_{dm}(l) = f_m(x)|_{x \in J_{dm,l}}$. Then we move up the tree $\mathfrak{T}$ and associate every non-leaf node $J_{k,l}$ with $\Psi^\vee_{k,l}$,$\Psi^\wedge_{k,l}$,$W^\vee_{k,l}$ and $W^\wedge_{k,l}$, calculated according to (14)-(17). $W^\vee_{k,l}$ and $W^\wedge_{k,l}$ are the Haar morphological wavelet coefficients of f_m computed on the fixed dyadic structure $\mathfrak{T}$. Notice that these are also implicitly dependent on $\mathfrak{T}$ although $\mathfrak{T}$ is not in the subscript.

With this setup, we now establish the following generalizations of Theorems 1 and 2 in higher dimensional spaces. Theorems 3 and 4 are not generalized because there is no known extension of the Besov norm in higher dimensional spaces that is dependent on the dyadic decomposition order.

Theorem 6.

$$E_{m,\mathfrak{T}}(f_m) = \sum_{k=0}^{dm-1} \sum_{l=0}^{2^k-1} (2\alpha_{k+1} - \alpha_k) d_{k,l}(f_m).$$

Theorem 7.

$$E_{m,\mathfrak{T}}(f_m) = \sum_{k=0}^{dm-1} \sum_{l=0}^{2^k-1} \alpha_{k+1} \left(|W^\vee_{k,l}| + |W^\wedge_{k,l}| \right).$$

To prove these two theorems, one can simply transform f_m to the 1D signal $\Psi_{dm}(l) = f_m(x)|_{x \in J_{dm,l}}$ and then apply the 1D results (Theorem 1 and Theorem 2) to Ψ_{dm}.

B. Choosing the Best Adaptive Dyadic Structure

For any signal f_m, all results in Section VI-A hold for any fixed $\mathfrak{T}$. Now we ask a new question: How do we choose the best dyadic structure $\mathfrak{T}$? From a signal representation perspective, a good $\mathfrak{T}$ should yield simple representations of the sublevel sets of f_m, i.e., the complexity measure $E_{m,\mathfrak{T}}(f_m)$ will be lower if the structure of $\mathfrak{T}$ is better adapted to the structure of f_m. Hence we seek:

$$\mathfrak{T}_{opt} \in \arg\min_{\mathfrak{T}} \{E_{m,\mathfrak{T}}(f_m)\}. \tag{30}$$

We can solve this problem by dynamic programming. The basic idea is to search for the optimum splitting recursively in a bottom-up fashion. For a hyperrectangle B in a full dyadic tree at depth k_B, define a "**sub-partition tree**" $\mathfrak{T}_B$ as any binary tree with root node B that satisfies properties **P2** and **P3** in Section VI-A. If k indexes the depth in the full dyadic tree, then $s(k) = k - k_B$ is the depth in the sub-partition tree $\mathfrak{T}_B$. The morphological Haar wavelet coefficients $W_{s(k),l}$ of f_m are calculated on $\mathfrak{T}_B$. Define the "**partial loss**" of $\mathfrak{T}_B$ as:

$$L(\mathfrak{T}_B) = \sum_{k=k_B}^{dm-1} \sum_{l=0}^{2^{s(k)}-1} \alpha_{k+1} \left(|W^\vee_{s(k),l}| + |W^\wedge_{s(k),l}| \right). \tag{31}$$

with $L(\mathfrak{T}_B)$ defined to be 0 for $k_B = dm$.

The dynamic programming algorithm recursively solves for the optimal sub-partition tree $\mathfrak{T}_{B,opt} \in \arg\min_{\mathfrak{T}_B} L(\mathfrak{T}_B)$ and stores the solutions in dictionary $\mathcal{D}_{k_B}$ as entry:

$$a(B) = \left(\max_{x \in B} f_m, \ \min_{x \in B} f_m, \ \mathfrak{T}_{B,opt}, \ L(\mathfrak{T}_{B,opt}) \right).$$

For leaf nodes B, $\mathfrak{T}_{B,opt}$ is a single node B and $L(\mathfrak{T}_B) = 0$. This trivial solution is stored into dictionary $\mathcal{D}_{dm}$. For a non-leaf node B, assume the color of $\mathfrak{T}_B$'s root node is c, this

```
For all leaves $B$ store the following in the dictionary $\mathcal{D}_{dm}$:
    $a(B) = (f_m(x)|_{x\in B},\ f_m(x)|_{x\in B},\ \{B\},\ 0)$.
for $k = dm-1, \dots, 1, 0$ do
    Initialize $\mathcal{D}_k = \phi$.
    for all $a(U) \in \mathcal{D}_{k+1}$ do
        for $c = 1, 2, \dots, d$ do
            if The $c$-th dimensional side length of $U$ is 1 then
                continue
            end if
            $V \leftarrow$ The unique hyperrectangle that could be the
            sibling of $U$ under a parent node $B$ of color $c$.
            $B \leftarrow U \cup V$.
            Retrieve entries $a(U)$ and $a(V)$ from $\mathcal{D}_{k+1}$:
                $a(U) = (\max_U,\ \min_U,\ \mathfrak{T}_{U,opt},\ L(\mathfrak{T}_{U,opt}))$,
                $a(V) = (\max_V,\ \min_V,\ \mathfrak{T}_{V,opt},\ L(\mathfrak{T}_{V,opt}))$.
            $\max_B \leftarrow \max\{\max_U, \max_V\}$,
            $\min_B \leftarrow \min\{\min_U, \min_V\}$,
            $\mathfrak{T}_{B,opt} \leftarrow \{\mathfrak{T}_{U,opt} \leftarrowtail B \rightarrowtail \mathfrak{T}_{V,opt}\}$,
            $L(\mathfrak{T}_{B,opt}) \leftarrow g(a(U), a(V))$
            Retrieve the existing entry for $B$ from $\mathcal{D}_k$:
                $a'(B) = \big(\max'_B,\ \min'_B,\ \mathfrak{T}'_{B,opt},\ L(\mathfrak{T}'_{B,opt})\big)$.
            if $a'(B) = \phi$ or $L(\mathfrak{T}_{B,opt}) < L(\mathfrak{T}'_{B,opt})$ then
                Replace $a'(B)$ with the entry:
                 $a(B) = (\max_B,\ \min_B,\ \mathfrak{T}_{B,opt},\ L(\mathfrak{T}_{B,opt}))$.
            end if
        end for
    end for
end for
Retrieve $a([0,1)^d)$ from $\mathcal{D}_0$ and output $\mathfrak{T}_{[0,1)^d,opt}$.
```

Fig. 2. Dynamic programming algorithm to solve (30).

cuts B into hyperrectangles B_U^c and B_V^c. Connecting the sub-partition trees rooted at B_U^c and B_V^c with new root node B yields tree $\mathfrak{T}_B$. We represent this as $\mathfrak{T}_B = \{\mathfrak{T}_{B_U^c} \leftarrowtail B \rightarrowtail \mathfrak{T}_{B_V^c}\}$. $L(\mathfrak{T}_B)$ can be recursively calculated as:

$$\begin{aligned} &L(\mathfrak{T}_B) = L(\mathfrak{T}_{B_U^c}) + L(\mathfrak{T}_{B_V^c}) + \\ &\alpha_{k_B+1}(|\max_{B_U^c} f_m - \max_{B_V^c} f_m| + |\min_{B_U^c} f_m - \min_{B_V^c} f_m|). \end{aligned} \tag{32}$$

The RHS of (32) can be calculated using just the entries $a(B_U^c)$ and $a(B_V^c)$ in $\mathcal{D}_{k_B+1}$. So we denote the RHS of (32) as $g(a(B_U^c), a(B_V^c))$. Now we can solve for $L(\mathfrak{T}_{B,opt})$ by minimizing (32) over c:

$$L(\mathfrak{T}_{B,opt}) = \min_{c=1,2,\dots,d} g(a(B_U^c), a(B_V^c)).$$

Using this approach, once we have completed the dictionary $\mathcal{D}_{k_B+1}$, the sub-partition trees for all hyperrectangles at depth k_B can be easily computed. Thus we can search for the best sub-partition trees using a bottom-up dynamic programming algorithm. This is described in detail in Figure 2.

To analyze the complexity of this algorithm, we note that $\sum_{k=0}^{dm} |\mathcal{D}_k| = (2^{m+1}-1)^d$. This holds since any hyperrectangle is the direct product of d 1D intervals, for each of which there are $2^{m+1}-1$ distinct choices. The dictionary $\mathcal{D}_k$ can be implemented so that the search and insert operations are $\mathcal{O}(\log|\mathcal{D}_k|)$. Hence the time complexity of the algorithm is:

$$|\mathcal{D}_{dm}|\log|\mathcal{D}_{dm}| + d\sum_{k=0}^{dm-1} |\mathcal{D}_{k+1}|C(\log|\mathcal{D}_k| + \log|\mathcal{D}_{k+1}|)$$

$$< dC'(\sum_{k=0}^{dm}|\mathcal{D}_k|)\log(\sum_{k=0}^{dm}|\mathcal{D}_k|) = \mathcal{O}(md^2 2^d 2^{md}).$$

Assume it takes b bits to represent a float number. Then each entry of $\mathcal{D}_k$ takes $\log(2^{m+1}-1)^d + b + b + \log d^{dm} + b$ bits to store. Hence the space complexity of the algorithm is:

$$(\log(2^{m+1}-1)^d + \log d^{dm} + 3*b)\sum_{k=0}^{dm}|\mathcal{D}_k|$$

$$<C(d(m+1)+dm\log d)(2^{m+1}-1)^d < \mathcal{O}(md\log d 2^d 2^{md}).$$

From these expressions we deduce that if f_m is a 2D image with N pixels, then both complexities are $\mathcal{O}(N\log N)$.

C. A Joint Denoising Algorithm Framework

Finally, we consider the joint optimization problem:

$$(\hat{f}, \hat{\mathfrak{T}}) \in \arg\min_{(f,\mathfrak{T})} \left\{ ||f - \tilde{f}||_2^2 + \lambda E_{m,\mathfrak{T}}(f) \right\}, \tag{33}$$

where $\lambda > 0$ is a regularization coefficient. In this formulation, we want to denoise the signal $\tilde{f}$ while jointly determining the dyadic structure best adapted to the denoised signal. Hence the regularizer $E_{m,\mathfrak{T}}(f)$ controls the complexity of the denoised signal under *its best* representation. This is a complex problem due to the non-linearity of $E_{m,\mathfrak{T}}(f)$ and the combinatorial choices of $\mathfrak{T}$. To solve the problem, we propose to iterate between optimizing $\hat{\mathfrak{T}}$ with $\hat{f}$ fixed and $\hat{f}$ with $\hat{\mathfrak{T}}$ fixed:

$$\text{T step: } \hat{\mathfrak{T}} \ \leftarrow\ \arg\min_{\mathfrak{T}} \left\{ ||\hat{f} - \tilde{f}||_2^2 + \lambda E_{m,\mathfrak{T}}(\hat{f}) \right\}, \tag{34}$$

$$\text{F step: } \hat{f} \ \leftarrow\ \arg\min_{f} \left\{ ||f - \tilde{f}||_2^2 + \lambda E_{m,\hat{\mathfrak{T}}}(f) \right\}. \tag{35}$$

(34) can be solved using the dynamic programming algorithm discussed in Section VI-B. (35) can be solved using the SOCP method discussed in Section V. Acting together these algorithms seek to iteratively solve (33).

VII. Experimental Results

While this paper is focused on extending and making connections between various signal regularization frameworks, we nevertheless feel it is important to examine the empirical performance of the denoising scheme that results. We do so using two experiments.

In the first experiment, we use a synthetic image to explore the effect of the regularizer $E_{m,\mathfrak{T}}(f)$ under different s and $\mathfrak{T}$ on a typical image denoising scenario in which we are given a noisy observation $\tilde{f}$ (PSNR=24dB) of the synthetic image (Figure 3 top panel). For three different images f: 1) the original signal, 2) an edge smoothed version of the original signal, and 3) the noisy observation, we plot $\log(\|f - \tilde{f}\|_2^2 + 4E_{m,\mathfrak{T}}(f)) - 4s$ as s ranges over the interval $[0,1]$ and $\mathfrak{T}$ ranges over three dyadic structures. The three dyadic structures $\mathfrak{T}$ are: the vertical decomposition (first decompose into single

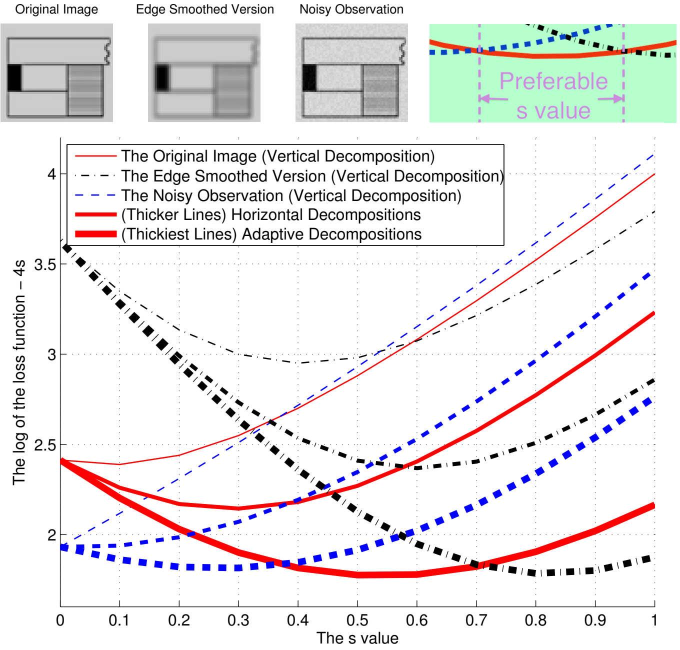
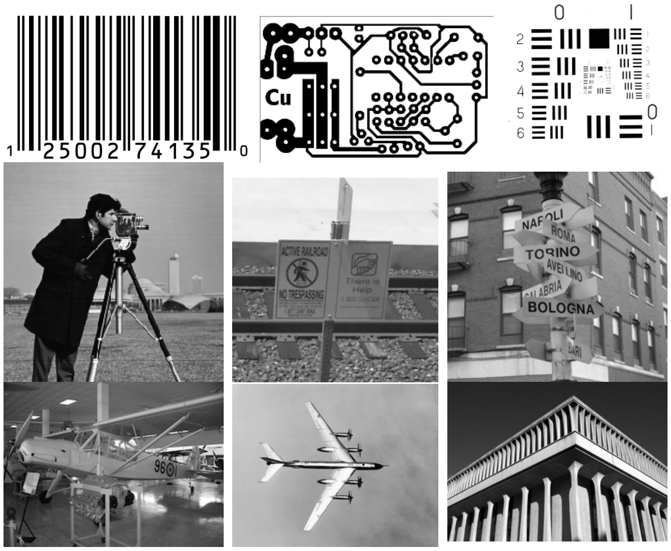


Fig. 3. The image denoising loss function $\|f-\tilde{f}\|_2^2+\lambda E_{m,\mathfrak{T}}(f)$ $(\lambda=4)$ for three images f, three $\mathfrak{T}$ and $s\in(0,1]$. This demonstrates the importance of choosing an appropriate s and using an adaptive dyadic structure $\mathfrak{T}$.

columns), the horizontal decomposition (first decompose into single rows), and the adaptive decomposition that minimizes $E_{m,\mathfrak{T}}(f)$. In Figure 3, groups of curves displayed with the same line style correspond to the same image f; groups of curves displayed with the same line width correspond to the same $\mathfrak{T}$. The pairwise intersections of the three curves for the same $\mathfrak{T}$ (same line width) form a triangular region. For the values of s defining the lower leg of this triangular region, the original signal has the lowest total loss. However, if s is too large, the loss function prefers the edge smoothed image. This is why conventional algorithms ($s=1$) blur edge features. On the other hand, if s is too small, $E_{m,\mathfrak{T}}(f)$ allows complex trees and the loss function prefers the noisy observation. The results also show that using adaptive dyadic structures yields a better signal representation. The complexity measure $E_{m,\mathfrak{T}}(f)$ is reduced when the dyadic structure $\mathfrak{T}$ is changed from vertical to horizontal and then to adaptive. Also, the aforementioned triangle has the largest base and the largest area when using the adaptive dyadic structure, suggesting a wider range of preferable s and more robust edge preservation.

In a second experiment, we evaluate our algorithm's denoising performance in terms of PSNR on 9 test images in Figure 4. Each test image is corrupted with Gaussian noise (input PSNR = 10dB). First we compare the performance of various wavelet methods. These use spatially local information (at various scales) to accomplish denoising. Specifically, we compare: (A) 2D wavelet hard thresholding (2DWh). (B) 1D wavelet hard thresholding (Wh). To do this, we first transform the image into an 1D signal using a fixed dyadic structure $\mathfrak{T}$. We tried 3 fixed $\mathfrak{T}$ denoted by (c), (r) and (m) [(c): columnwise decomposition, (r) row-wise decomposition, and (m): a mixed column-row alternating decomposition.] (C) 2D wavelet soft thresholding (2DWs). (D) 1D wavelet soft thresholding (Ws), similar to (B). (E) A non-adaptive formulation (35), where $\mathfrak{T}$ is selected to be the best of the three choices in (B). (F) Our proposed joint denoising formulation (33). For these methods, we average the denoising results over cycle spinnings



ID	2DWh	Wh ($\mathfrak{T}$)	(35) ($\mathfrak{T}$)	(33) (s)	TV	ISKR	BM3D-σ
1	19.9	20.2 (c)	20.4 (c)	24.8 (0.8)	18.9	21.4	27.1
2	16.8	16.2 (m)	16.9 (m)	18.1 (0.7)	17.5	17.5	19.7
3	20.4	19.2 (m)	19.2 (m)	22.1 (0.9)	20.3	20.7	23.7
4	15.5	15.6 (m)	16.1 (m)	16.1 (0.9)	15.8	15.4	15.8
5	22.4	22.2 (m)	23.0 (m)	23.9 (0.9)	23.2	23.3	24.2
6	20.6	20.2 (m)	21.2 (m)	21.7 (0.9)	21.3	21.8	22.2
7	21.1	20.7 (m)	21.6 (m)	21.9 (0.9)	21.7	22.1	22.6
8	23.7	23.4 (m)	23.4 (m)	25.2 (0.9)	24.5	25.7	27.1
9	19.9	19.0 (m)	19.8 (m)	21.0 (0.9)	20.2	21.9	24.1

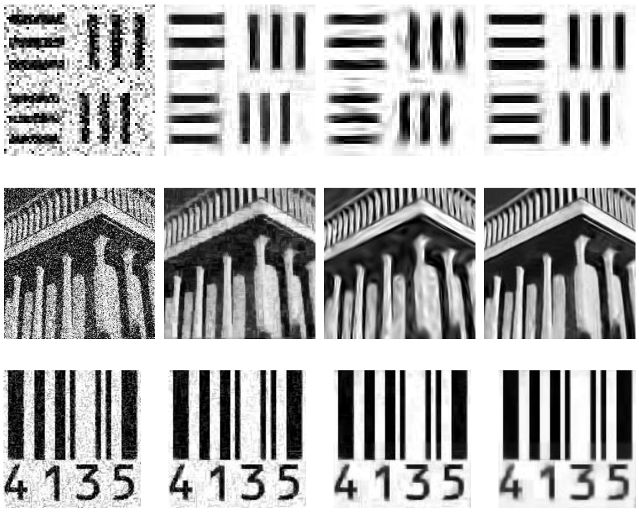


Fig. 4. The denoising performance of different algorithms on 9 test images. The original images are numbered from 1 (top left) to 9 (bottom right). Each image is corrupted with Gaussian noise with input PSNR=10dB. The PSNR (in dB) of each algorithm's output is shown in the table, along with the best fixed dyadic structure $\mathfrak{T}$ and parameter s. Soft thresholding algorithms (C),(D) are always worse than their counterparts in hard thresholding (A),(B) and therefore are not shown in the table. Below the table, in the first and second row we show the denoising results from patches in image 3 and 9. From left to right we show: the noisy image, denoised images using our method (33), ISKR and BM3D-σ. In the third row we show the denoising results on a patch of image 1 with $s=0.6,0.7,0.8,0.9$ in our formulation (33) respectively.

[30] of shifts $(i, j), 0 \leq i, j \leq 3$.

For benchmark purposes, we also compared with: (G) Total variation (TV) using the formulation in [1] and a recent fast algorithm in [31]; (H) Iterative steering kernel regression (ISKR) [32] (these two methods are also spatially local algorithms); and (I) Block matching 3D with known σ (BM3D-σ). BM3D [33] is a very competitive spatially-nonlocal algorithm and it consistently matches the performance of recent new algorithms such as [34]. For BM3D-σ we give the BM3D algorithm the true noise level σ. We expect BM3D-σ to outperform state-of-the-art spatially local methods and to outperform BM3D for which the σ is not known and has to be estimated.

For all methods we grid-search the parameters (including wavelet types from db1(Haar) to db5 for (A)-(D), global smoothing parameter and the number of iterations for (H)) and report the best PSNR.

The results in Figure 4 are very encouraging. In terms of PSNR numbers, our method (33) has the best performance among all the wavelet-based algorithms and exhibits competitive performance even when compared to the best state-of-the-art denoising algorithms. In terms of visual results, our method offers clear edge preservation and minimum edge artifacts. However, as expected the nonlocal block matching algorithm BM3D-σ yields superior results because of its nonlocal nature and the abundance of homogenous regions throughout the image (see the first two demonstrated image patches). We see the importance of using an *adaptive* dyadic structure $\mathfrak{T}$, since (33) can yield improved PSNR compared to (35). Moreover, the best s selected by the grid search is lower for images with more edges (1 and 2). As a specific example, the visual illustration at the end of Figure 4 shows that for image 1, $s = 0.8$ yields the best denoising result. Larger s blurs the edges and smaller s tolerates too much noise. This confirms our previous analysis on the meaning of s, and supports the hypotheses in [18] that it's advantageous to use unbalanced trees to represent sublevel sets in images with salient edges. The parameter s conveniently controls the level of edge preservation, but tuning s could result in additional computation time.

To denoise the first of the three patches shown in Figure 4 on an Intel Xeon X5570 2.93GHz processor, (A)-(D) take 0.02s per cycle spinning. (G) takes 0.02s, (H) takes 8.54s and (I) takes 0.05s. Our algorithm (F) takes 4.09s per cycle spinning, in which the bottle neck is the 4.02 seconds (98%) spent by the external SOCP solver SeDuMi [35] to solve (35). At the cost of a small reduction in performance, we can circumvent this computation burden by replacing morphological Haar wavelets with Haar wavelets. This results in at least 50X speed up - this idea is explored in [36]. Morphological wavelets are used here for consistency with the theoretical derivations. All the computation times are reported under the optimal parameters and do not include the time required for searching for these parameters (e.g. the s in our method). The average number of iterations ((34),(35)) is 2.1.

VIII. Conclusion

We have introduced a new image regularizer $E_{m,\mathfrak{T}}(f)$ which measures the complexity of a signal f based on the complexity of the dyadic decision trees required to represent the sublevel sets of f. We established important connections of $E_{m,\mathfrak{T}}(f)$ to classification trees, morphological wavelets, the wavelet shrinkage method, dynamic ranges, and Besov spaces. These connections help address a central question: How to control edge preservation in image denoising? We have shown that the parameter $s \in [0, 1]$ in our formulation is the key factor that controls edge preservation. A smaller s makes the regularizer more tolerant to edge discontinuities. We have shown that s is connected to the preference of unbalanced tree structures in machine learning, to the use of level-dependent thresholding in wavelet denoising, and to the scope of the Besov space $B^s_{1,1}$ as the underlying smoothness space. While the traditional CART algorithm and conventional wavelet shrinkage correspond to the single point $s = 1$, we demonstrated (theoretically and empirically) that using other values of s offers advantages in edge preservation and image denoising. $E_{m,\mathfrak{T}}(f)$ is dependent on the dyadic decomposition structure $\mathfrak{T}$ and we presented efficient algorithms to jointly estimate the signal and its best adaptive dyadic structure. This provides a theoretical basis to adaptively determine a dyadic wavelet decomposition structure for dimensions greater than 1. Experiments show that using an adaptive dyadic structure can improve denoising performance.

The proposed approach outperforms all existing Haar wavelet thresholding algorithms that don't use adaptive trees. By design, our method uses spatially-local nonlinear averaging. Not surprisingly, state-of-the-art denoising methods employing joint, spatially-nonlocal averaging yield superior performance.

Acknowledgements

The authors thank Prof. Ingrid Daubechies for fruitful discussions on wavelets and Besov space. Zhen James Xiang gratefully acknowledges the support of Princeton University through a Charlotte Elizabeth Procter honorific fellowship.

Appendix A
Examination of Condition B

Let T_{k+1} be the balanced tree with 2^{k+1} leaf nodes ($k \geq 1$). Let T' be formed by a local modification of T_{k+1} as follows: merge two leaf nodes in T into a size 2^{-k} node, and split another leaf node in T into two size $2^{-(k+2)}$ nodes. So T_{k+1} and T' have the same number of leaf nodes. We say that $\{\alpha_k\}$ prefers unbalanced trees if for all $k \geq 1$, T_{k+1} has a higher complexity than T', i.e., no balanced tree is a local stationary point of the complexity measure. By the construction of T', $\{\alpha_k\}$ prefers unbalanced trees if and only if for each $k \geq 1$:

$$\begin{aligned} & 2^{k+1}\alpha_{k+1} > (2^{k+1} - 3)\alpha_{k+1} + 2\alpha_{k+2} + \alpha_k, \\ \Leftrightarrow \quad & 2\alpha_{k+1} - \alpha_k \ > \ 2\alpha_{k+2} - \alpha_{k+1}, \\ \Leftrightarrow \quad & \beta_{k+1} \ > \ \beta_{k+2} \end{aligned}$$

where $\beta_k = 2\alpha_k - \alpha_{k-1}$ with $\beta_k > 0$ (by Condition A), for $k \geq 2$. Hence $\beta_k < c\gamma^k$, $k \geq 2$, for some constants $c > 0, \gamma \in (0, 1)$ is sufficient to ensue that $\{\alpha_k\}$ prefers unbalanced trees. Moreover, since $\beta_k = 2\alpha_k - \alpha_{k-1}$, Condition B is sufficient to ensure that $\beta_k < c\gamma^k$, $k \geq 2$, for some constants $c > 0, \gamma \in (0,1)$. Hence Condition B is sufficient to ensure that $\{\alpha_k\}$ prefers unbalanced trees.

Appendix B
Proof of Theorem 1

By the definition in Section III-A and $\alpha_0 = 0$:

$$\Phi(T_\gamma^m) = \sum_{k=1}^{m}\sum_{l=0}^{2^k-1} \alpha_k [\![I_{k,l} \text{ is a leaf node in } T_\gamma^m]\!]. \qquad (36)$$

According to the rule of constructing T_γ to represent S_γ described in Section III-B, the necessary and sufficient conditions for $I_{k,l}$ ($k \geq 1$) to be a leaf node are:

1) Interval $I_{k-1,\lfloor \frac{l}{2} \rfloor}$ (the parent node of $I_{k,l}$) is *not* entirely contained in S_γ nor in $[0,1)\backslash S_\gamma$. *And:*
2) Either $k = m$, or that interval $I_{k,l}$ is entirely contained in S_γ or in $[0,1)\backslash S_\gamma$.

Since $S_\gamma = \{x \in [0,1) : f(x) \leq \gamma\}$, an interval I being not entirely contained in S_γ nor in $[0,1)\backslash S_\gamma$ is equivalent to $\gamma \in \{\min_I\} \cup (\inf_I, \sup_I)$. Here $\min_I$ is our short hand notation for $\min_{x\in I} f(x)$. The similar notation also applies to $\inf_I$ and $\sup_I$. $\{\min_I\}$ is defined to be an empty set if f doesn't have an attainable minimum value on I. Because $I_{k,l} \subset I_{k-1,\lfloor \frac{l}{2} \rfloor}$, $\gamma \in \{\min_{I_{k,l}}\} \cup (\inf_{I_{k,l}}, \sup_{I_{k,l}}) \Rightarrow \gamma \in \{\min_{I_{k-1,\lfloor \frac{l}{2} \rfloor}}\} \cup (\inf_{I_{k-1,\lfloor \frac{l}{2} \rfloor}}, \sup_{I_{k-1,\lfloor \frac{l}{2} \rfloor}})$. Using these arguments we can express the indicator function of whether $I_{k,l}$ is a leaf node, $[\![I_{k,l}$ is a leaf node in $T_\gamma^m]\!]$, using

$$[\![\gamma \in \{\min_{I_{k-1,\lfloor \frac{l}{2} \rfloor}}\} \cup (\inf_{I_{k-1,\lfloor \frac{l}{2} \rfloor}}, \sup_{I_{k-1,\lfloor \frac{l}{2} \rfloor}})]\!] - [\![\gamma \in \{\min_{I_{k,l}}\} \cup (\inf_{I_{k,l}}, \sup_{I_{k,l}})]\!]$$

when $1 \leq k < m$, and using

$$[\![\gamma \in \{\min_{I_{k-1,\lfloor \frac{l}{2} \rfloor}}\} \cup (\inf_{I_{k-1,\lfloor \frac{l}{2} \rfloor}}, \sup_{I_{k-1,\lfloor \frac{l}{2} \rfloor}})]\!]$$

when $k = m$. Plugging these into (36):

$$\begin{aligned}
\Phi(T_\gamma^m) &= \sum_{k=1}^{m}\sum_{l=0}^{2^k-1} \alpha_k [\![\gamma \in \{\min_{I_{k-1,\lfloor \frac{l}{2} \rfloor}}\} \cup (\inf_{I_{k-1,\lfloor \frac{l}{2} \rfloor}}, \sup_{I_{k-1,\lfloor \frac{l}{2} \rfloor}})]\!] \\
&\quad - \sum_{k=1}^{m-1}\sum_{l=0}^{2^k-1} \alpha_k [\![\gamma \in \{\min_{I_{k,l}}\} \cup (\inf_{I_{k,l}}, \sup_{I_{k,l}})]\!] \\
&= \sum_{k=0}^{m-1}\sum_{l=0}^{2^{k+1}-1} \alpha_{k+1} [\![\gamma \in \{\min_{I_{k,\lfloor \frac{l}{2} \rfloor}}\} \cup (\inf_{I_{k,\lfloor \frac{l}{2} \rfloor}}, \sup_{I_{k,\lfloor \frac{l}{2} \rfloor}})]\!] \\
&\quad - \sum_{k=0}^{m-1}\sum_{l=0}^{2^k-1} \alpha_k [\![\gamma \in \{\min_{I_{k,l}}\} \cup (\inf_{I_{k,l}}, \sup_{I_{k,l}})]\!] \\
&= \sum_{k=0}^{m-1}\sum_{l=0}^{2^k-1} (2\alpha_{k+1} - \alpha_k) [\![\gamma \in \{\min_{I_{k,l}}\} \cup (\inf_{I_{k,l}}, \sup_{I_{k,l}})]\!].
\end{aligned}$$

Integrating this equation over γ and using

$$\int [\![\gamma \in \{\min_{I_{k,l}}\} \cup (\inf_{I_{k,l}}, \sup_{I_{k,l}})]\!] d\gamma = d_{k,l}$$

gives us equation (11) in Theorem 1. □

Appendix C
Proof of Corollary 1

The sufficiency part is obvious using the convexity of $\sup$ and $-\inf$. For necessity, constructing the following signals:

$$f_1(x) = \begin{cases} 1 & \text{if } x \in \cup_{j=1,2,\ldots,2^k} I_{k+1,2j-1}, \\ 0 & \text{otherwise.} \end{cases}$$

$$f_2(x) = \begin{cases} 1 & \text{if } x \in I_{k+1,0} \cup (\cup_{j=2,3,\ldots,2^k} I_{k+1,2j-1}), \\ 0 & \text{otherwise.} \end{cases}$$

The average of the two functions $\bar{f}(x) = \frac{1}{2}(f_1(x) + f_2(x))$ is:

$$\bar{f}(x) = \begin{cases} \frac{1}{2} & \text{if } x \in I_{k+1,0} \cup I_{k+1,1}, \\ 1 & \text{if } x \in I_{k+1,2j-1}, j = 2,3,\ldots,2^k, \\ 0 & \text{otherwise.} \end{cases}$$

If $E_m(\cdot)$ is convex, then we have $2E_m(\bar{f}) < E_m(f_1) + E_m(f_2)$, which yields $2\alpha_{k+1} - \alpha_k > 0$. □

Appendix D
Proof of Theorem 2

We use $d_{k,l}$ as a shorthand notation for $d_{k,l}(f_m)$. By definition (14)- (17) and the fact that f_m is constant on $I_{m,l}$:

$$\Psi_k^{\vee}(l) = \max_{x \in I_{k,l}} f_m(x) = \sup_{x \in I_{k,l}} f_m(x)$$
$$\Psi_k^{\wedge}(l) = \min_{x \in I_{k,l}} f_m(x) = \inf_{x \in I_{k,l}} f_m(x)$$

$$\begin{aligned}
d_{k,l} &= \sup_{x \in I_{k,l}} f_m(x) - \inf_{x \in I_{k,l}} f_m(x) = \Psi_k^{\vee}(l) - \Psi_k^{\wedge}(l) \\
&= \Psi_{k+1}^{\vee}(2l) \vee \Psi_{k+1}^{\vee}(2l+1) - \Psi_{k+1}^{\wedge}(2l) \wedge \Psi_{k+1}^{\wedge}(2l+1) \\
&= \frac{1}{2}(\Psi_{k+1}^{\vee}(2l) + \Psi_{k+1}^{\vee}(2l+1) + |\Psi_{k+1}^{\vee}(2l) - \Psi_{k+1}^{\vee}(2l+1)|) \\
&\quad - \frac{1}{2}(\Psi_{k+1}^{\wedge}(2l) + \Psi_{k+1}^{\wedge}(2l+1) - |\Psi_{k+1}^{\wedge}(2l) - \Psi_{k+1}^{\wedge}(2l+1)|) \\
&= \frac{1}{2} d_{k+1,2l} + \frac{1}{2} d_{k+1,2l+1} + \frac{1}{2}\left(|W_{k,l}^{\vee}| + |W_{k,l}^{\wedge}|\right).
\end{aligned}$$

$$\Rightarrow \quad 2d_{k,l} - d_{k+1,2l} - d_{k+1,2l+1} = |W_{k,l}^{\vee}| + |W_{k,l}^{\wedge}|. \qquad (37)$$

On the other hand, starting from Theorem 1, we have:

$$\begin{aligned}
E_m(f_m) &= \sum_{k=0}^{m-1}\sum_{l=0}^{2^k-1} 2\alpha_{k+1} d_{k,l} - \sum_{k=0}^{m}\sum_{l=0}^{2^k-1} \alpha_k d_{k,l} \\
&= \sum_{k=0}^{m-1}\sum_{l=0}^{2^k-1} 2\alpha_{k+1} d_{k,l} - \sum_{k=0}^{m-1}\sum_{l=0}^{2^{k+1}-1} \alpha_{k+1} d_{k+1,l}
\end{aligned}$$

($k \leftarrow k+1$ on the 2nd term and using $\alpha_0 = 0, d_{m,l} = 0$)

$$\begin{aligned}
&= \sum_{k=0}^{m-1}\sum_{l=0}^{2^k-1} \alpha_{k+1}(2d_{k,l} - d_{k+1,2l} - d_{k+1,2l+1}) \\
&= \sum_{k=0}^{m-1}\sum_{l=0}^{2^k-1} \alpha_{k+1}\left(|W_{k,l}^{\vee}| + |W_{k,l}^{\wedge}|\right) \text{ (Using (37)).} \quad □
\end{aligned}$$

Appendix E
Proof of Theorem 3

We only need to establish the following bound:

$$C_1 \sum_{j=0}^{m-1} 2^{sj} w_1(f, 2^{-j})_{\mathcal{L}^1} \leq E_m(f)$$

Because if this is proved, then taking $m \to \infty$ yields the conclusion of the theorem. To prove this bound, we first prove that for a fixed j:

$$w_1(f, 2^{-j})_{\mathcal{L}^1} \leq 4 \sum_{k=0}^{j} \sum_{l=0}^{2^k-1} 2^{-j} d_{k,l}. \tag{38}$$

Since $w_1(f, 2^{-j})_{\mathcal{L}^1}$ is upper-bounded by $d_{0,0}$, (38) holds for $j = 0, 1, 2$. For $j \geq 3$, divide $[0, 1)$ into these intervals:

$$H_{k,l} = \left[\frac{l+\frac{1}{2}}{2^k} - \frac{1}{2^j}, \frac{l+1/2}{2^k} + \frac{1}{2^j}\right), K = [0, \frac{1}{2^j}) \cup [1-\frac{1}{2^j}, 1),$$

where $k = 0, 1, \ldots, j-2, l = 0, 1, \ldots, 2^k - 1$. These intervals are not overlapping. Consider an arbitrary h with $|h| \leq 2^{-j}$. For all $x \in H_{k,l}$, both x and $x - h$ are in the interval $I_{k,l}$. So

$$\begin{aligned}
&\|\Delta_h^1 f\|_{\mathcal{L}^p(\Omega_{h,1})} = \int_{\max(h,0)}^{\min(1+h,1)} |f(x) - f(x-h)| dx \\
\leq& \sum_{k=0}^{j-2} \sum_{l=0}^{2^k-1} \int_{H_{k,l}} |f(x) - f(x-h)| dx + \int_K |f(x) - f(x-h)| dx \\
\leq& \sum_{k=0}^{j-2} \sum_{l=0}^{2^k-1} |H_{k,l}| d_{k,l} + |K| d_{0,0} \\
=& \sum_{k=0}^{j-2} \sum_{l=0}^{2^k-1} 2 \times 2^{-j} d_{k,l} + 2 \times 2^{-j} d_{0,0} \leq 4 \sum_{k=0}^{j} \sum_{l=0}^{2^k-1} 2^{-j} d_{k,l}.
\end{aligned}$$

This is true for any $|h| \leq 2^{-j}$. Therefore taking the supremum $\sup_{|h| \leq 2^{-j}}$ proves (38). Now summing over j on (38):

$$\begin{aligned}
&\sum_{j=0}^{m-1} 2^{sj} w_1(f, 2^{-j})_{\mathcal{L}^1} \leq 4 \sum_{j=0}^{m-1} \sum_{k=0}^{j} \sum_{l=0}^{2^k-1} 2^{sj} 2^{-j} d_{k,l} \\
=& 4 \sum_{k=0}^{m-1} \sum_{j=k}^{m-1} \sum_{l=0}^{2^k-1} 2^{-(1-s)j} d_{k,l} \leq 4 \sum_{k=0}^{m-1} \sum_{j=k}^{\infty} \sum_{l=0}^{2^k-1} 2^{-(1-s)j} d_{k,l} \\
=& 4 \sum_{k=0}^{m-1} \sum_{l=0}^{2^k-1} \frac{2^{-(1-s)k} d_{k,l}}{1 - 2^{-(1-s)}} = \frac{1}{C_1} \sum_{k=0}^{m-1} \sum_{l=0}^{2^k-1} (2\alpha_{k+1} - \alpha_k) d_{k,l}.
\end{aligned}$$

This proves the bound with $C_1 = \frac{(2^s-1)(2-2^s)}{4 \times 2}$. □

Appendix F
Proof of Theorem 4

Because $f_{\mathcal{X}_m}(x)$ is a sampled version of f, we have $d_{k,l}(f_{\mathcal{X}_m}) \leq d_{k,l}(f)$ for any $I_{k,l}$. Using Theorem 1 we have:

$$\forall m, \mathcal{X}_m : \quad E_m(f_{\mathcal{X}_m}) \leq E_m(f). \tag{39}$$

On the other hand, for any given m, the evaluation of $E_m(f) = \sum_{k=0}^{m-1} \sum_{l=0}^{2^k-1} (2\alpha_{k+1} - \alpha_k) d_{k,l}(f)$ depends on all the $2(2^m - 1)$ supremum and infimum values of f on intervals $I_{k,l}$. For any given ε, we can choose $2(2^m - 1)$ corresponding anchor points so that the function value on each point is within $\frac{\varepsilon}{2(2^m-1)}$ of the corresponding supremum/infimum value. There exists an integer n so that interval $I_{n,l}$ contains at most one such anchor point. Therefore we can choose sample points set $\tilde{\mathcal{X}}_n$ to contain the anchor points. So we have

$$\forall m, \varepsilon : \exists n \text{ and } \tilde{\mathcal{X}}_n, \text{ s.t.: } E_n(f_{\tilde{\mathcal{X}}}) \geq E_m(f_{\tilde{\mathcal{X}}}) \geq E_m(f) - \varepsilon.$$

This, combined with (39), yields (20).

To prove (21) under σ-Hölder continuity, we will first prove:

$$\forall m, \mathcal{X}_m, \varepsilon : \ E_m(f_{\mathcal{X}_m}) \geq E_m(f) - \varepsilon - 4C \times 2^{(s-\sigma)m}. \tag{40}$$

This is because for any interval $I_{k,l}$ we can find $x_1, x_2 \in I_{k,l}$ s.t. $|f(x_1) - f(x_2)| \geq d_{k,l}(f) - \frac{\varepsilon}{2^m-1}$. Denote $x_1 \in I_{m,l_1}, x_2 \in I_{m,l_2}$ (l_1, l_2 could be equal). Let x_{l_1} and x_{l_2} be the two sample points in $\mathcal{X}_m$ that are in I_{m,l_1} and I_{m,l_2} respectively. We have:

$$\begin{aligned}
&d_{k,l}(f_{\mathcal{X}_m}) \geq |f_{\mathcal{X}_m}(x_{l_1}) - f_{\mathcal{X}_m}(x_{l_2})| = |f(x_{l_1}) - f(x_{l_2})| \\
\geq& |f(x_1) - f(x_2)| - |f(x_1) - f(x_{l_1})| - |f(x_2) - f(x_{l_2})| \\
\geq& d_{k,l}(f) - \frac{\varepsilon}{2^m - 1} - C|x_1 - x_{l_1}|^\sigma - C|x_2 - x_{l_2}|^\sigma \\
\geq& d_{k,l}(f) - \frac{\varepsilon}{2^m - 1} - 2C2^{-\sigma m}.
\end{aligned}$$

Taking the summation $\sum_{k=0}^{m-1} \sum_{l=0}^{2^k-1} (2\alpha_{k+1} - \alpha_k)$ yields (40). Taking the limit on (40), combined with (39), yields (21). □

References

[1] L. Rudin, S. Osher, and E. Fatemi, "Nonlinear total variation based noise removal algorithms," *Physica D*, vol. 60, no. 1-4, pp. 259–268, 1992.

[2] S. Mallat, *A wavelet tour of signal processing*. Academic, San Diego, CA, 1998.

[3] D. Donoho and J. Johnstone, "Ideal spatial adaptation by wavelet shrinkage," *Biometrika*, vol. 81, no. 3, p. 425, 1994.

[4] D. Donoho and I. Johnstone, "Adapting to unknown smoothness via wavelet shrinkage." *Journal of the American Statistical Association*, vol. 90, no. 432, 1995.

[5] W. Hardle, G. Kerkyacharian, D. Picard, and A. Tsybakov, *Wavelets, Approximation, and Statistical Applications. Lecture Notes in Statistics*. Springer (New York), 1997.

[6] A. Cohen, I. Daubechies, O. Guleryuz, and M. Orchard, "On the importance of combining wavelet-based nonlinear approximation with coding strategies," *IEEE Transactions on Information Theory*, vol. 48, no. 7, pp. 1895–1921, 2002.

[7] A. Chambolle, R. DeVore, N. Lee, and B. Lucier, "Nonlinear wavelet image processing: Variational problems, compression, and noise removal through wavelet shrinkage," *IEEE Transactions on Image Processing*, vol. 7, no. 3, pp. 319–335, 1998.

[8] H. Choi and R. Baraniuk, "Wavelet statistical models and Besov spaces," *Lecture Notes In Statistics*, pp. 9–30, 2003.

[9] S. Murthy, "Automatic construction of decision trees from data: A multi-disciplinary survey," *Data Mining and Knowledge Discovery*, vol. 2, no. 4, pp. 345–389, 1998.

[10] L. Breiman, J. Friedman, R. Olshen, and C. Stone, *Classification and regression trees. 1984*. Wadsworth, Belmont, 1984.

[11] J. Quinlan, *C4.5: programs for machine learning*. Morgan Kaufmann, 2003.

[12] C. Scott, "Dyadic decision trees," Ph.D. dissertation, University of Wisconsin, 2004.

[13] C. Scott and R. Nowak, "Dyadic classification trees via structural risk minimization," in *Advances in Neural Information Processing Systems*, 2002, pp. 375–382.

[14] ——, "Near-minimax optimal classification with dyadic classification trees," in *Advances in Neural Information Processing Systems*, vol. 16, 2003.

[15] ——, "Minimax-optimal classification with dyadic decision trees," *IEEE Transactions on Information Theory*, vol. 52, no. 4, pp. 1335–1353, 2006.

[16] J. Goutsias and H. Heijmans, "Nonlinear multiresolution signal decomposition schemes-Part I: Morphological pyramids," *IEEE Transactions on Image Processing*, vol. 9, no. 11, pp. 1862–1876, 2000.

[17] H. Heijmans and J. Goutsias, "Nonlinear multiresolution signal decomposition schemes-Part II: Morphological wavelets," *IEEE Transactions on Image Processing*, vol. 9, no. 11, pp. 1897–1913, 2000.

[18] R. Willett and R. Nowak, "Minimax optimal level-set estimation," *IEEE Transactions on Image Processing*, vol. 16, no. 12, pp. 2965–2979, 2007.

[19] Z. Harmany, R. Willett, A. Singh, and R. Nowak, "Controlling the error in fMRI: Hypothesis testing or set estimation?" in *5th IEEE International Symposium on Biomedical Imaging*, 2008, pp. 552–555.

[20] Z. Xiang and P. Ramadge, "Morphological wavelets and the complexity of dyadic trees," in *IEEE International Conference on Acoustics, Speech and Signal Processing*, 2010.

[21] ——, "Morphological wavelet transform with adaptive dyadic structures," in *IEEE International Conference on Image Processing*, 2010.

[22] D. Donoho, "Nonlinear wavelet methods for recovery of signals, densities, and spectra from indirect and noisy data," *Proceedings of Symposia in Applied Mathematics*, 1993.

[23] H. Triebel, *Theory of function spaces*. Springer Basel, 2010.

[24] U. Amato, A. Antoniadis, and M. Pensky, "Wavelet kernel penalized estimation for non-equispaced design regression," *Statistics and Computing*, vol. 16, no. 1, pp. 37–55, 2006.

[25] S. Loustau, "Penalized empirical risk minimization over Besov spaces," *Electronic Journal of Statistics*, 2009.

[26] V. Vapnik and S. Kotz, *Estimation of dependences based on empirical data*. Springer-Verlag New York Inc, 2006.

[27] G. Lugosi and K. Zeger, "Concept learning using complexity regularization," *IEEE Transactions on Information Theory*, vol. 42, no. 1, pp. 48–54, 1996.

[28] L. Mason, P. Bartlett, and M. Golea, "Generalization error of combined classifiers," *Journal of Computer and System Sciences*, vol. 65, no. 2, pp. 415–438, 2002.

[29] L. Reyzin and R. Schapire, "How boosting the margin can also boost classifier complexity," in *International Conference on Machine Learning*, 2006, pp. 753–760.

[30] R. Coifman and D. Donoho, "Translation-invariant de-noising," *Lecture Notes in Statistics*, pp. 125–125, 1995.

[31] A. Chambolle, "An algorithm for total variation minimization and applications," *Journal of Mathematical Imaging and Vision*, vol. 20, no. 1, pp. 89–97, 2004.

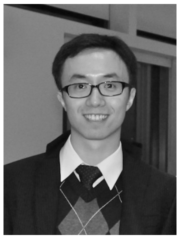
[32] H. Takeda, H. Takeda, S. Member, S. Farsiu, P. Milanfar, and S. Member, "Kernel regression for image processing and reconstruction," *IEEE Transactions on Image Processing*, vol. 16, pp. 349–366, 2007.

[33] K. Dabov, A. Foi, V. Katkovnik, and K. Egiazarian, "Image denoising by sparse 3-D transform-domain collaborative filtering," *IEEE Transactions on Image Processing*, vol. 16, no. 8, pp. 2080 –2095, aug. 2007.

[34] W. Dong, X. Li, L. Zhang, and G. Shi, "Sparsity-based image denoising vis dictionary learning and structural clustering," in *IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 2011.

[35] J. Sturm, "SeDuMi, Advanced Optimization Laboratory, McMaster University," 2009.

[36] Z. J. Xiang and P. J. Ramadge, "Learning a wavelet tree for multichannel image denoising," in *IEEE International Conference on Image Processing*, 2011.



Zhen James Xiang received his B.E. from the Department of Electrical Engineering, Tsinghua University, China in 2007. He is currently a Ph.D. candidate in the Department of Electrical Engineering, Princeton University. During his Ph.D. study, he worked at NEC Labs America Inc. at Cupertino, CA as a summer intern in 2010.

Mr. Xiang is a recipient of the Princeton Charlotte Elizabeth Procter honorific fellowship and the Princeton Francis Robin Upton fellowship. He was also awarded the gold medal of the International Mathematics Olympiad in 2003.

Peter J. Ramadge received the B.Sc. and B.E. degrees and the M.E. degree from the University of Newcastle, Australia, and the Ph.D. degree from the Department of Electrical Engineering at the University of Toronto, Canada. He joined the faculty of Princeton University in September 1984, where he is currently Gordon Y. S. Wu Professor of Engineering, Professor of Electrical Engineering and Chair of the Department of Electrical Engineering. He has been a visiting Professor at the Massachusetts Institute of Technology and a Visiting Research Scientist at IBM's Tokyo Research Laboratory. He is a Fellow of the IEEE and a member of SIAM.

Prof. Ramadge has received several honors and awards including: a paper selected for inclusion in IEEE book Control Theory: Twenty Five Seminal Papers (1932-1981); an Outstanding Paper Award from the Control Systems Society of the IEEE; the Walter C. Johnson Prize for Teaching Excellence from the Department of Electrical Engineering, Princeton University; the Convocation Medal for Professional Excellence from the University of Newcastle, Australia; an NCR Corporation Award for Excellence in Teaching; an IBM Faculty Development Award; and the University Medal from Newcastle University, Australia. His current research interests are in computational signal processing and machine learning with applications to large scale data including video, images, sound, and functional magnetic resonance imaging.