A Box-Spline Framework for Inverse Problems With Continuous-Domain Sparsity Constraints

Mehrsa Pourya, *Graduate Student Member, IEEE*, Aleix Boquet-Pujadas, and Michael Unser, *Fellow, IEEE*

***Abstract*—The formulation of inverse problems in the continuum eliminates discretization errors and allows for the exact incorporation of priors. In this paper, we formulate a continuous-domain inverse problem over a search space of continuous and piecewise-linear functions parameterized by box splines. We present a numerical framework to solve those inverse problems with total variation (TV) or its Hessian-based extension (HTV) as regularizers. We show that the box-spline basis allows for exact and efficient convolution-based expressions for both TV and HTV. Our optimization strategy relies on a multiresolution scheme whereby we progressively refine the solution until its cost stabilizes. We test our framework on linear inverse problems and demonstrate its ability to effectively reach a stage beyond which the refinement of the search space no longer decreases the optimization cost.**

***Index Terms*—Continuous and piecewise linear, discretization, total variation, Hessian total variation, multiresolution.**

I. Introduction

THE recovery of an unknown signal from a set of noisy measurements—an inverse problem—plays a central role in the field of imaging [1], [2]. The classic solution of inverse problems involves an arbitrary discretization step whereby one expresses the unknown signal in some basis. The pixel basis is popular for its simplicity. Using such a basis, one formulates the inverse problem as

$$\arg\min_{\boldsymbol{c}\in\mathbb{R}^N} \left(\mathrm{E}(\mathbf{H}\boldsymbol{c}, \boldsymbol{y}) + \lambda\mathcal{R}(\boldsymbol{c})\right), \tag{1}$$

where the discrete measurements are $\boldsymbol{y} = (y_m)_{m=1}^M \in \mathbb{C}^M$ and $\boldsymbol{c} = (c_n)_{n=1}^N \in \mathbb{R}^N$ are the pixel values. In this formulation, $\mathbf{H} \in \mathbb{C}^{M\times N}$ denotes the discretized forward operator (the system matrix), which models the acquisition process. The loss functional $\mathrm{E} : \mathbb{C}^M \times \mathbb{C}^M \to \mathbb{R}^+$ quantifies data fidelity. The regularizer $\mathcal{R} : \mathbb{R}^N \to \mathbb{R}^+$ incorporates prior knowledge about the unknown signal to ensure the well-posedness of the problem. The hyperparameter $\lambda \in \mathbb{R}^+$ balances the data fidelity and regularization terms. There is extensive literature devoted to the solution of (1), especially with sparsity-promoting regularizers supported by the theory of compressed sensing [3], [4], [5], [6], [7], [8], [9].

A. Continuous-Domain Inverse Problems

Although the discrete formulation (1) is widespread, a formulation of the inverse problem in the continuum offers several advantages. The continuous formulation eliminates discretization errors and allows for the direct employment of regularizers in the continuum. Moreover, it matches the underlying signal better and allows for its evaluation at any resolution. These reasons have motivated many studies for the understanding of continuous-domain inverse problems [10], [11], [12], [13], [14], [15], [16], [17], [18].

To pose an inverse problem in the continuum, one denotes the unknown signal as the d-dimensional function $f : \mathbb{R}^d \to \mathbb{R}$ and formulates the problem as

$$\arg\min_{f\in\mathcal{X}} \left(\mathrm{E}(\boldsymbol{\nu}(f), \boldsymbol{y}) + \lambda\mathcal{R}(f)\right), \tag{2}$$

where $\mathcal{X}$ represents an adequate search space of functions and $\boldsymbol{\nu} : \mathcal{X} \to \mathbb{C}^M$ denotes the forward operator. Concretely, $\boldsymbol{\nu}(f) = (\nu_m(f))_{m=1}^M$ with $\nu_m : \mathcal{X} \to \mathbb{C}$ representing the effect of the mth detector of the imaging system. The regularizer $\mathcal{R} : \mathcal{X} \to \mathbb{R}^+$ acts in the continuum on the function $f \in \mathcal{X}$.

To make Problem (2) tractable, one needs to identify a suitable search space $\mathcal{X}$. A natural choice for $\mathcal{X}$ comes from the finiteness of the regularization term. This induces a native space $\mathcal{B}_\mathcal{R}$, which is *intuitively* the largest function space where $\mathcal{R}(f)$ is well-defined [19], [20]. In the best-case scenario, the solutions of (2) for $\mathcal{X} = \mathcal{B}_\mathcal{R}$ are characterized using a representer theorem [21], [22], [23]. This allows one to represent the solution with a finite number of parameters, turning the problem into a computationally feasible one. Representer theorems depend on the regularizer and often identify solutions through the extreme points of the regularization ball [24]. However, one cannot always rely on representer theorems for discretization as the solutions they provide sometimes take intractable forms. This urges the need for alternative strategies to tackle those cases.

In this paper, we focus on regularizers such as total variation (TV) and Hessian total variation (HTV), as expressed in the continuum [25], [26]. The pixel-based implementation of these continuously defined regularizers provides strong tools for the solution of inverse problems through compressed-sensing methods [27], [28], [29].

Manuscript received 16 October 2023; revised 2 April 2024; accepted 11 May 2024. Date of publication 17 May 2024; date of current version 30 May 2024. This work was supported in part by the European Research Council (ERC Project FunLearn) under Grant 101020573, in part by the Swiss National Science Foundation under Grant 200020_219356, and in part by Sinergia under Grant CRSII5 198569. The associate editor coordinating the review of this manuscript and approving it for publication was Prof. Alin M Achim. *(Corresponding author: Mehrsa Pourya.)*

The authors are with the Biomedical Imaging Group, École polytechnique fédérale de Lausanne, 1015 Lausanne, Switzerland (e-mail: mehrsa.pourya@epfl.ch; aleix.boquetipujadas@epfl.ch; michael.unser@epfl.ch).

This article has supplementary downloadable material available at https://doi.org/10.1109/TCI.2024.3402376, provided by the authors.

Digital Object Identifier 10.1109/TCI.2024.3402376

2333-9403 © 2024 IEEE. Personal use is permitted, but republication/redistribution requires IEEE permission.
See https://www.ieee.org/publications/rights/index.html for more information.

Authorized licensed use limited to: EPFL LAUSANNE. Downloaded on June 25,2024 at 10:28:57 UTC from IEEE Xplore. Restrictions apply.

The continuous-domain inverse problems over $\mathcal{B}_{\mathrm{TV}}$ and $\mathcal{B}_{\mathrm{HTV}}$ have been studied theoretically. In essence, TV regularization promotes piecewise-constant solutions [22], [30], [31]. Likewise, HTV regularization promotes continuous and piecewise-linear (CPWL) solutions [32], [33]. Algorithms in the one-dimensional case (d = 1) exist [34], [35], [36]. They rely on B-spline-based parameterizations of the signals on uniform grids [37]. In this paper, we generalize such frameworks to multiple dimensions.

B. Contributions

We develop a framework to solve the continuous-domain problem (2) with TV and HTV regularizations in multiple dimensions. We restrict the search space $\mathcal{X}$ to the set of CPWL functions defined on uniform grids, and construct our framework through three main contributions.

(i) Development of a CPWL search space in arbitrary dimension: To generate a CPWL search space, we use first-order box splines on a Cartesian grid. Contrarily to their (more widespread) tensor-product (of linear B-splines) cousins [38], box splines guarantee a CPWL mapping, regardless of the dimensionality and without any compromise in approximation quality [39], [40], [41], [42], [43], [44], [45]. In Section III, we present our CPWL basis and derive its properties.

(ii) Exact computations of continuous-domain TV and HTV: We show that the use of first-order box splines allows for the exact computation of TV and HTV through efficient convolutions (see Theorems 2 and 3). Our continuous-domain computations eliminate the need for singular-value decomposition (SVD) in HTV, which is the computational bottleneck of the pixel-based approaches. This efficiency carries on to proximal-based optimization.

(iii) Multiresolution scheme for optimization: We follow a multiresolution scheme to find a proper grid size for the proposed CPWL model. This scheme progressively adjusts the grid size until the cost of optimization stabilizes. It relies on multiscale extensions of our CPWL search space with dyadic grid sizes. We prove that the resulting multiscale search spaces are refinable (see Proposition 1). We present the details of the discretization at each scale and the numerical solvers in Sections IV and V, respectively.

We validate our framework in Section VI, where we present numerical results for the reconstruction of continuous-domain signals from Fourier measurements. We observe that the optimization cost stabilizes with the proper decay of the stepsize of the grid, which indicates that further refinement of the search space is not productive. Moreover, our proposed multiresolution strategy accelerates the optimization of the problem on fine grids. We also discuss the reconstruction quality and computational complexity of our CPWL-based discretizations of TV and HTV, as compared to their fully discrete counterparts. Notably, we observe that our approach accelerates the computations for HTV regularization.

C. Related Works

A classic approach to the solution of problems with a functional model of the underlying signal is finite-element discretization [46], [47]. The authors of [48] deployed a finite-element discretization on a uniform triangulation to solve a TV-regularized inverse problem. The CPWL parameterization of [48] is similar to that of our box-spline-based approach, which partitions the domain into the simplices of the Kuhn-Freudenthal triangulation. However, the method in [48] is specific to denoising problems with a continuum of measurements. In contrast, we consider a more realistic imaging setup wherein a finite set of measurements is obtained through a forward operator $\boldsymbol{\nu}$. Our spline-based formulation offers several advantages. It allows for closed-form expressions of various forward operators, in combination with TV and HTV regularization. Further, it results in an exact and convolutional upsampling scheme to refine the grid in any dimension.

Works like [49], [50] view TV regularization from the perspective of partial differential equations. They find explicit forms of the solution for denoising problems with a continuum of measurements.

Gridless (or off-the-grid) algorithms provide another alternative to the continuous-domain solution of TV regularization [51]. Indeed, the solution of the off-the-grid algorithms is modeled as a sum of indicator functions over simple sets. Although such approaches are mathematically elegant and work well for images with simple structures, they do not seem to generalize well to natural images [52]. They also rely on heavy optimization techniques. By contrast, multiresolution schemes reduce the artifacts caused by the approximation grid and benefit from the efficient solvers developed for discrete optimization.

II. Mathematical Preliminaries

A. Continuous and Piecewise-Linear Functions

A function $f : \mathbb{R}^d \to \mathbb{R}$ is continuous and piecewise-linear (CPWL) if

1) it forms a continuous map;
2) its domain $\mathbb{R}^d$ can be partitioned into a set $\{P_k\}$ of non-overlapping polytopes such that $\mathbb{R}^d = \bigcup P_k$ and

$$f(\boldsymbol{x}) = \sum_k (\boldsymbol{a}_k^\top \boldsymbol{x} + b_k) \mathbb{1}_{P_k}(\boldsymbol{x}) \tag{3}$$

for some $\boldsymbol{a}_k \in \mathbb{R}^d$ and $b_k \in \mathbb{R}$. The indicator function $\mathbb{1}_{P_k} : \mathbb{R}^d \to \{0, 1\}$ maps to 1 when evaluated inside the polytope P_k, and to 0 elsewhere.

By inspection of (3), the gradient of a CPWL function f is

$$\boldsymbol{\nabla} f = \sum_k \boldsymbol{a}_k \mathbb{1}_{P_k}(\boldsymbol{x}). \tag{4}$$

Note that the continuity of f at the boundaries of the polytopes imposes some non-trivial constraints on $\{\boldsymbol{a}_k\}$ and $\{b_k\}$ in (3) and (4). Therefore, arbitrary choices of such parameters are not allowed.

B. Box Splines

A box spline $B_{\boldsymbol{\Xi}_p} : \mathbb{R}^d \to \mathbb{R}$ of nonnegative integer order p in the space $\mathbb{R}^d$ is defined through vectors $\boldsymbol{\xi}_r \in \mathbb{R}^d$, $r \in [1, \ldots, d+p]$. They are gathered into a matrix $\boldsymbol{\Xi}_p = [\boldsymbol{\xi}_1 \ \cdots \ \boldsymbol{\xi}_{d+p}] \in \mathbb{R}^{d \times (d+p)}$ [53]. The vectors in the set

Authorized licensed use limited to: EPFL LAUSANNE. Downloaded on June 25,2024 at 10:28:57 UTC from IEEE Xplore. Restrictions apply.

$\{\boldsymbol{\xi}_r\}_{r=1}^d$ are linearly independent, meaning that $\boldsymbol{\Xi}_0$ is full-rank. The piecewise-constant box spline $B_{\boldsymbol{\Xi}_0}$ is defined as

$$B_{\boldsymbol{\Xi}_0}(\boldsymbol{x}) = \begin{cases} \frac{1}{|\det \boldsymbol{\Xi}_0|}, & \boldsymbol{x} = \sum_{r=1}^d t_r \boldsymbol{\xi}_r \text{ for } t_r \in [0,1] \\ 0, & \text{otherwise.} \end{cases} \tag{5}$$

For $p \geq 1$, the following recursion holds:

$$B_{\boldsymbol{\Xi}_p}(\boldsymbol{x}) = \int_0^1 B_{\boldsymbol{\Xi}_{p-1}}(\boldsymbol{x} - t\boldsymbol{\xi}_{d+p}) \mathrm{d}t. \tag{6}$$

As we shall see, this implies that the first-order box spline $B_{\boldsymbol{\Xi}_1}$ is a CPWL function, irrespective of the dimension d.

C. Freudenthal-Kuhn Triangulation

A simplex is the convex hull of $d+1$ affinely independent points in $\mathbb{R}^d$. A triangulation of $\mathbb{R}^d$ is its partition into simplices with disjoint interiors.

The Freudenthal-Kuhn triangulation partitions $\mathbb{R}^d$ into the T-cubes $\{[Tk_1, T(k_1+1)] \times \cdots \times [Tk_d, T(k_d+1)]\}_{\boldsymbol{k} \in \mathbb{Z}^d}$, where $\boldsymbol{k} = (k_r)_{r=1}^d$ and $T > 0$ is the grid size. Each $T-$cube is further subdivided into simplices $S_{\pi_q, \boldsymbol{k}, T}$ that are uniquely determined by the set $\{\boldsymbol{v}_0, \ldots, \boldsymbol{v}_d\}$ of their vertices, where

$$\boldsymbol{v}_0 = T\boldsymbol{k}, \tag{7}$$

$$\boldsymbol{v}_r = \boldsymbol{v}_{r-1} + T\boldsymbol{e}_{\pi_q(r)}, \quad r \in 1, \ldots, d. \tag{8}$$

Here, π_q is the qth member of Π—the set of all permutations of $\{1, \ldots, d\}$—and $\pi_q(r)$ denotes the rth element of this permutation [54]. The elements of $\{\boldsymbol{e}_1, \ldots, \boldsymbol{e}_d\}$ are the canonical bases of $\mathbb{R}^d$. An example of the Freudenthal-Kuhn triangulation for $d = 2$ is illustrated in Fig. 1(a).

D. Total Variation

The isotropic total variation (TV) of a differentiable function $f : \mathbb{R}^d \to \mathbb{R}$ is defined as

$$\mathrm{TV}(f) = \int_{\mathbb{R}^d} \|\boldsymbol{\nabla} f(\boldsymbol{x})\|_2 \, \mathrm{d}\boldsymbol{x}, \tag{9}$$

where $\boldsymbol{\nabla} f(\boldsymbol{x})$ denotes the gradient of f. The TV regularization promotes piecewise-constant solutions whose gradient magnitude $\|\boldsymbol{\nabla} f\|_2$ is zero almost everywhere. A more general definition of TV that accommodates non-differentiable functions is provided in Appendix A.

E. Hessian Total Variation

The Hessian total variation (HTV) is a generalization of TV that involves second-order differentials. For a twice-differentiable function, we write

$$\begin{aligned} \mathrm{HTV}(f) &= \int_{\mathbb{R}^d} \|\mathbf{H}_f(\boldsymbol{x})\|_{\mathcal{S}_1} \mathrm{d}\boldsymbol{x} \\ &= \int_{\mathbb{R}^d} \sum_{r=1}^d \sigma_r(\mathbf{H}_f(\boldsymbol{x})) \mathrm{d}\boldsymbol{x}, \end{aligned} \tag{10}$$

where $\mathbf{H}_f(\boldsymbol{x})$ is the Hessian matrix of f at location $\boldsymbol{x}$ and $\|\cdot\|_{\mathcal{S}_1}$ is the matrix Schatten-one norm. The latter is given by the ℓ_1 norm of the singular values of a matrix denoted by $\sigma_r(\cdot)$. Definition (10) does not encompass CPWL functions; however, one can generalize it to accommodate such functions as

$$\mathrm{HTV}(f) = \|\mathbf{H}_f\|_{\mathcal{S}_1, \mathcal{M}}, \tag{11}$$

where $\mathbf{H}_f$ is the Hessian of f in the generalized sense of distribution. Here, $\|\cdot\|_{\mathcal{S}_1, \mathcal{M}}$ denotes the Schatten-total-variation mixed norm. Intuitively, HTV is the total-variation norm ($\mathcal{M}$ norm) of the mapping $\boldsymbol{x} \to \|\mathbf{H}_f(\boldsymbol{x})\|_{\mathcal{S}_1}$ while the $\mathcal{M}$ norm is a generalization of the L_1 norm (see [26] for more details). The null space of the HTV seminorm consists of affine mappings, and CPWL functions are dense in the ball of the HTV seminorm, including its extremal points [32]. The HTV regularization favors CPWL functions with few affine pieces.

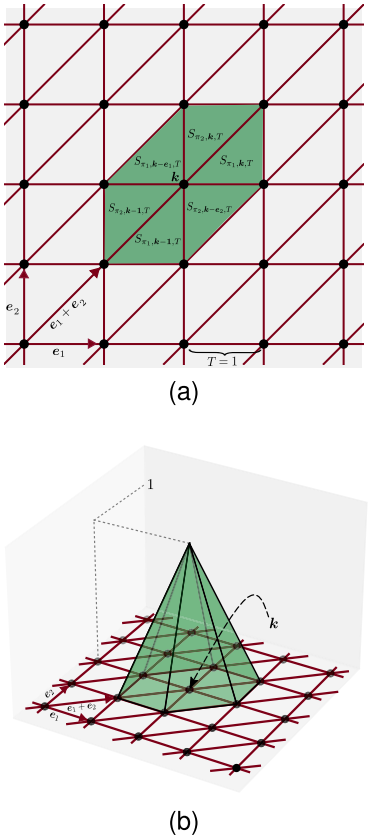

Fig. 1. (a) An example of the Freudenthal-Kuhn triangulation in the two-dimensional space: $\pi_1 = (1,2)$, $\pi_2 = (2,1)$, and the identified simplices illustrate the support of a first-order box spline on the Cartesian grid. (b) The basis function $\varphi(\cdot - \boldsymbol{k})$ is a shifted box spline on the Cartesian grid that is positioned at point $\boldsymbol{k} \in \mathbb{R}^2$. Such functions are affine over each simplex of the Freudenthal-Kuhn triangulation.

III. Continuous-Domain Model and Calculations

In this section, we first define our multiscale CPWL search spaces and investigate their properties in Section III-A. We then compute their TV and HTV in Section III-B.

Authorized licensed use limited to: EPFL LAUSANNE. Downloaded on June 25,2024 at 10:28:57 UTC from IEEE Xplore. Restrictions apply.

A. Multiscale CPWL Search Spaces

We first derive some properties of box splines defined on the Cartesian grid. They are useful for the definition and analysis of our proposed search space. In particular, we build multiscale CPWL search spaces $\{\mathcal{X}_{(s)}|s\in\mathbb{Z}\}$ that are refinable (Proposition 1) to support our proposed multiresolution scheme. Our refinement is dyadic, with the grid size of $\mathcal{X}_{(s)}$ being $(s)=2^{-s}$.

1) Cartesian Box Splines: We set $\boldsymbol{\Xi}_0=\mathbf{I}$, where $\mathbf{I}$ is the identity matrix. From (5), we have that

$$B_{\boldsymbol{\Xi}_0}(\boldsymbol{x})=\prod_{r=1}^{d}\text{box}(x_r), \tag{12}$$

where $\boldsymbol{x}=(x_r)_{r=1}^d$ and the function box: $\mathbb{R}\to\mathbb{R}$ is defined as

$$\text{box}(x)=\begin{cases}1, & 0\le x\le 1\\ 0, & \text{otherwise.}\end{cases} \tag{13}$$

To define the piecewise-linear Cartesian box spline $B_{\boldsymbol{\Xi}_1}:\mathbb{R}^d\to\mathbb{R}$, we choose $\boldsymbol{\xi}_{d+1}=\sum_{r=1}^d \boldsymbol{e}_r=\mathbf{1}$. By (6), we have that

$$B_{\boldsymbol{\Xi}_1}(\boldsymbol{x})=\int_0^1 B_{\boldsymbol{\Xi}_0}(\boldsymbol{x}-t\boldsymbol{\xi}_{d+1})\mathrm{d}t=\int_0^1\prod_{r=1}^{d}\text{box}(x_r-t)\mathrm{d}t$$

$$=\int_{-\infty}^{+\infty}\text{box}(t)\prod_{r=1}^{d}\text{box}(x_r-t)\mathrm{d}t. \tag{14}$$

We then define our basis generator φ as

$$\varphi(\boldsymbol{x})=B_{\boldsymbol{\Xi}_1}(\boldsymbol{x}+\mathbf{1}), \tag{15}$$

where the shift by $\mathbf{1}$ centers the basis function around the origin. An example of the basis generator shifted to position $\boldsymbol{k}$ is shown in Fig. 1(b).

Theorem 1 (Basis-Function Properties): The piecewise-linear box spline $\varphi:\mathbb{R}^d\to\mathbb{R}$ defined by (15) has the following properties.

i) Explicit closed-form formula

$$\varphi(\boldsymbol{x})=(1+\min(x_1,\cdots,x_d,0)-\max(x_1,\cdots,x_d,0))_+, \tag{16}$$

where $\boldsymbol{x}=(x_r)_{r=1}^d$, and $(x)_+:=\max(x,0)$.

ii) Interpolatory on the Cartesian grid

$$\forall\boldsymbol{k}\in\mathbb{Z}^d:\quad \varphi(\boldsymbol{k})=\begin{cases}1, & \boldsymbol{k}=\mathbf{0}\\ 0, & \text{otherwise.}\end{cases} \tag{17}$$

iii) Explicit Fourier transform

$$\hat{\varphi}(\boldsymbol{\omega})=\frac{\mathrm{e}^{\mathrm{j}\mathbf{1}^\top\boldsymbol{\omega}}-1}{\mathrm{j}\mathbf{1}^\top\boldsymbol{\omega}}\prod_{r=1}^{d}\frac{1-\mathrm{e}^{-\mathrm{j}\omega_r}}{\mathrm{j}\omega_r}$$

$$=\text{sinc}\left(\frac{\mathbf{1}^\top\boldsymbol{\omega}}{2\pi}\right)\prod_{r=1}^{d}\text{sinc}\left(\frac{\omega_r}{2\pi}\right) \tag{18}$$

where $\boldsymbol{\omega}=(\omega_r)\in\mathbb{R}^d$ and $\text{sinc}(x)=\frac{\sin\pi x}{\pi x}$.

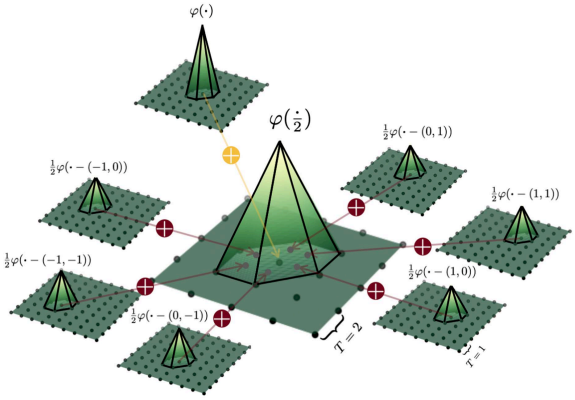

Fig. 2. Refinability of the box-spline basis function in the dimension $d=2$.

iv) Refinability

$$\varphi\left(\frac{\cdot}{2}\right)=\frac{1}{2}\sum_{\boldsymbol{k}\in\{0,1\}^d}(\varphi(\cdot+\boldsymbol{k})+\varphi(\cdot-\boldsymbol{k})). \tag{19}$$

Note that (16) expresses φ as the composition of two elementary CPWL functions. Since the CPWL property is preserved through composition [55], this automatically implies that φ is CPWL as well. In Fig. 2, we illustrate (19) for the two-dimensional case ($d=2$). Specifically, $\varphi(\frac{\cdot}{2})$, which is a basis generator for a grid of size $T=2$, can be constructed exactly by a linear combination of seven shifted versions of $\varphi(\cdot)$, which is a basis generator on a grid of size $T=1$. The weights are all 0.5 except for $\boldsymbol{k}=\mathbf{0}$ in (19), where $\varphi(\cdot)$ gets a double weight.

Proof of Theorem 1:

i) The geometric interpretation of (14) and (15) implies that

$$\varphi(\boldsymbol{x})=\text{Leb}\left(\bigcap_{r=1}^{d}\text{supp}_t\{\text{box}(x_r-t+1)\}\right.$$

$$\left.\bigcap\text{supp}_t\{\text{box}(t)\}\right), \tag{20}$$

where supp_t returns the support of the input function with respect to the variable t, and Leb denotes the Lebesgue measure. From (13),

$$\varphi(\boldsymbol{x})=\text{Leb}\left(\bigcap_{r=1}^{d}[x_r,x_r+1]\bigcap[0,1]\right). \tag{21}$$

Through Lemma 1 of Appendix[1] A, we have that

$$\varphi(\boldsymbol{x})=\begin{cases}\text{Leb}(C_d), & \text{Len}_{C_d}\ge 0\\ \text{Leb}(\emptyset), & \text{Len}_{C_d}<0,\end{cases} \tag{22}$$

where $C_d=[\max(x_1,\cdots,x_d,0),\min(x_1,\cdots,x_d,0)+1]$, and $\text{Len}_{C_d}=(1+\min(x_1,\cdots,x_d,0)-\max(x_1,\cdots,x_d,0))$. Therefore

$$\varphi(\boldsymbol{x})=\begin{cases}\text{Len}_{C_d}, & \text{Len}_{C_d}\ge 0\\ 0, & \text{Len}_{C_d}<0\end{cases} \tag{23}$$

[1]The appendix is provided in supplementary materials.

Authorized licensed use limited to: EPFL LAUSANNE. Downloaded on June 25,2024 at 10:28:57 UTC from IEEE Xplore. Restrictions apply.

implies that

$$\varphi(\boldsymbol{x}) = [\text{Len}_{C_d}]_+ \tag{24}$$

and completes the proof.

ii) One can verify this property through (16) and the fact that $(\max(k_1, \cdots, k_d, 0) - \min(k_1, \cdots, k_d, 0)) \geq 1$ when $\boldsymbol{k} = (k_r)_{r=1}^d \in \mathbb{Z}^d \setminus \{\boldsymbol{0}\}$.

iii) This result has been proven elsewhere.[2]

iv) Let us assume that $\varphi(\frac{\cdot}{2})$ can be expressed through a linear combination of the shifted versions of $\varphi(\cdot)$ as

$$\varphi\left(\frac{\cdot}{2}\right) = \sum_{\boldsymbol{k}\in\mathbb{Z}^d} u[\boldsymbol{k}]\varphi(\cdot - \boldsymbol{k}). \tag{25}$$

To find u such that (25) holds, we take the Fourier transform on both sides, which yields that

$$U(\mathrm{e}^{\mathrm{j}\boldsymbol{\omega}}) = \frac{2^d\hat{\varphi}(2\boldsymbol{\omega})}{\hat{\varphi}(\boldsymbol{\omega})}, \tag{26}$$

where $U(\mathrm{e}^{\mathrm{j}\boldsymbol{\omega}}) = \sum_{\boldsymbol{k}\in\mathbb{Z}^d} u[\boldsymbol{k}]\mathrm{e}^{-\mathrm{j}\boldsymbol{\omega}^\top\boldsymbol{k}}$ (discrete-time Fourier transform of u). By (18) and after some simplifications, we have that

$$U(\mathrm{e}^{\mathrm{j}\boldsymbol{\omega}}) = \frac{1}{2}(\mathrm{e}^{\mathrm{j}\mathbf{1}^\top\boldsymbol{\omega}} + 1)\prod_{r=1}^d (1 + \mathrm{e}^{-\mathrm{j}\omega_r}). \tag{27}$$

We define $V(\mathrm{e}^{\mathrm{j}\boldsymbol{\omega}}) = \prod_{r=1}^d (1 + \mathrm{e}^{-\mathrm{j}\omega_r})$ and invoke separability to compute its inverse discrete-time Fourier transform v as

$$v[\boldsymbol{k}'] = \prod_{r=1}^d (\delta[k_r'] + \delta[k_r' + 1]) = \sum_{\boldsymbol{k}\in\{0,1\}^d} \delta[\boldsymbol{k}' + \boldsymbol{k}]. \tag{28}$$

From the properties of the Fourier transform, we have that

$$\begin{aligned} u[\cdot] &= \frac{1}{2}(v + v[\cdot - \mathbf{1}]) \\ &= \frac{1}{2}\sum_{\boldsymbol{k}\in\{0,1\}^d} (\delta[\cdot + \boldsymbol{k}] + \delta[\cdot + \boldsymbol{k} - \mathbf{1}]) \\ &= \frac{1}{2}\sum_{\boldsymbol{k}\in\{0,1\}^d} (\delta[\cdot + \boldsymbol{k}] + \delta[\cdot - \boldsymbol{k}]), \end{aligned} \tag{29}$$

where $\delta[\cdot]$ represents the Kronecker delta. Equations (29) and (25) lead to (19), which completes the proof. □

2) Definition of the Search Space: Equipped with the box spline $\varphi : \mathbb{R}^d \to \mathbb{R}$, we now define the CPWL function search space

$$\mathcal{X}_T = \left\{ \sum_{\boldsymbol{k}\in\mathbb{Z}^d} c[\boldsymbol{k}]\varphi\left(\frac{\cdot}{T} - \boldsymbol{k}\right), c \in \ell_2(\mathbb{Z}^d) \right\}, \tag{30}$$

where $\ell_2(\mathbb{Z}^d)$ denotes the space of discrete signals with finite energy. Each function $f : \mathbb{R}^d \to \mathbb{R} \in \mathcal{X}_T$ can be written as

$$f(\cdot) = \sum_{\boldsymbol{k}\in\mathbb{Z}^d} c[\boldsymbol{k}]\varphi\left(\frac{\cdot}{T} - \boldsymbol{k}\right), \tag{31}$$

where $\{c[\boldsymbol{k}]\}_{\boldsymbol{k}\in\mathbb{Z}^d}$ is the set of its so-called expansion coefficients.

The use of the box splines induces the following properties.

i) The affine mapping $f : \mathbb{R}^d \to \mathbb{R}$ of form $f(\boldsymbol{x}) = \boldsymbol{a}^\top\boldsymbol{x} + b$ can be produced by $\sum_{\boldsymbol{k}\in\mathbb{Z}^d}(\frac{1}{T}\boldsymbol{a}^\top\boldsymbol{k} + b)\varphi(\frac{\cdot}{T} - \boldsymbol{k})$ for some $\boldsymbol{a} \in \mathbb{R}^d, b \in \mathbb{R}$.

ii) The space $\mathcal{X}_T$ has the capacity to approximate any twice-differentiable function [56], [57].

iii) The set $\{\varphi(\frac{\cdot}{T} - \boldsymbol{k})\}_{\boldsymbol{k}\in\mathbb{Z}^d}$ forms a Riesz basis, which guarantees a unique and stable link between any function $f \in \mathcal{X}_T$ and its expansion coefficients c [58].

iv) The linear regions of the function $f \in \mathcal{X}_T$ are exactly the set of the simplices of the Freudenthal-Kuhn triangulation with the grid size T. The sampled value of f at a vertex $T\boldsymbol{k}$, $\boldsymbol{k} \in \mathbb{Z}^d$, of the triangulation is $f(T\boldsymbol{k}) = c[\boldsymbol{k}]$. On each simplex, f can be identified by the only hyperplane that interpolates the function values at the vertices of that simplex [58].

3) Multiscale Search Spaces: We define a series of multiscale search spaces $\{\mathcal{X}_{(s)} = \mathcal{X}_{2^{-s}} | s \in \mathbb{Z}\}$. At scale s, $\mathcal{X}_{(s)}$ satisfies (30) with $T = 2^{-s}$. Therefore, any $f_s : \mathbb{R}^d \to \mathbb{R} \in \mathcal{X}_{(s)}$ is given by

$$f_s(\cdot) = \sum_{\boldsymbol{k}\in\mathbb{Z}^d} c_s[\boldsymbol{k}]\varphi\left(2^s\cdot - \boldsymbol{k}\right) \tag{32}$$

where $\{c_s[\boldsymbol{k}]\}_{\boldsymbol{k}\in\mathbb{Z}^d}$ is the set of expansion coefficients of f_s at scale s.

Proposition 1 (Refinable Search Space): If $f \in \mathcal{X}_{(s)}$ with expansion coefficients c_s, then f is exactly representable in $\mathcal{X}_{(s+1)}$ with expansion coefficients

$$c_{s+1}[\cdot] = \sum_{\boldsymbol{k}\in\mathbb{Z}^d} c_s[\boldsymbol{k}]u[\cdot - 2\boldsymbol{k}], \tag{33}$$

where u is defined in (29).

Proof: If $f \in \mathcal{X}_{(s)}$, from (32), (25), and (29) we have that

$$\begin{aligned} f(\cdot) &= \sum_{\boldsymbol{k}\in\mathbb{Z}^d} c_s[\boldsymbol{k}] \sum_{\boldsymbol{k}'\in\mathbb{Z}^d} u[\boldsymbol{k}']\varphi(2^{s+1}\cdot - 2\boldsymbol{k} - \boldsymbol{k}') \\ &= \sum_{\boldsymbol{k}\in\mathbb{Z}^d}\sum_{\boldsymbol{k}'\in\mathbb{Z}^d} c_s[\boldsymbol{k}]u[\boldsymbol{k}' - 2\boldsymbol{k}]\varphi(2^{s+1}\cdot - \boldsymbol{k}'). \end{aligned} \tag{34}$$

This implies that

$$c_{s+1}[\boldsymbol{k}'] = \sum_{\boldsymbol{k}\in\mathbb{Z}^d} c_s[\boldsymbol{k}]u[\boldsymbol{k}' - 2\boldsymbol{k}] \tag{35}$$

and completes the proof. □

As a result of Theorem 1, we obtain the embedding $\cdots \subset \mathcal{X}_{(s-1)} \subset \mathcal{X}_{(s)} \subset \mathcal{X}_{(s+1)} \subset \cdots$. This means that any function at a coarse scale $s_1 \leq s$ is exactly representable at any finer scale s.

B. Exact and Efficient Computation of the Regularization

Here, we show that the TV and HTV seminorms of $f \in \mathcal{X}_T$ can be expressed exactly through efficient convolutions. Without loss of generality, our results are applicable to $f_s \in \mathcal{X}_{(s)}$.

[2] https://doi.org/10.1016/j.acha.2023.101581

Authorized licensed use limited to: EPFL LAUSANNE. Downloaded on June 25,2024 at 10:28:57 UTC from IEEE Xplore. Restrictions apply.

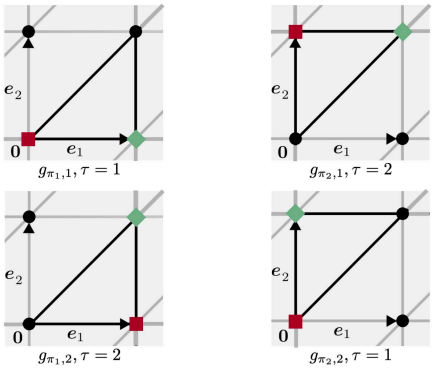

Fig. 3. Discrete filters $\{g_{\pi_q,r} | q, r \in \{1,2\}\}$ and the internal variable τ, as used for the calculation of gradient and TV in $d = 2$. The filters are zero everywhere except at the diamond and square points that correspond to the values of $+1$ and (-1), respectively. At each plot, the black boundaries identify the simplex over which the rth component of the gradient is computable through the depicted finite difference.

1) Total Variation: We first calculate the gradient of $f \in \mathcal{X}_T$ in Lemma 1; we then compute its TV seminorm in Theorem 2.

Lemma 1 (Gradient Calculation): The gradient of the function $f \in \mathcal{X}_T$ on the interior of simplices of its domain is

$$\nabla f(\cdot) = \frac{1}{T} \sum_{\boldsymbol{k} \in \mathbb{Z}^d} \sum_{q=1}^{d!} \sum_{r=1}^{d} \left(c * g_{\pi_q,r}\right)[\boldsymbol{k}] \mathbb{1}_{\pi_q,\boldsymbol{k},T}(\cdot) \boldsymbol{e}_r, \quad (36)$$

where $*$ denotes the discrete convolution operator, and $\mathbb{1}_{\pi_q,\boldsymbol{k},T} : \mathbb{R}^d \to \{0,1\}$ is the indicator function corresponding to the simplex $S_{\pi_q,\boldsymbol{k},T}$ of the Kuhn-Freudental triangulation. Moreover, we have that

$$g_{\pi_q,r}[\cdot] = \delta[\cdot + \sum_{n=1}^{\tau-1} \boldsymbol{e}_{\pi_q(n)} + \boldsymbol{e}_r] - \delta[\cdot + \sum_{n=1}^{\tau-1} \boldsymbol{e}_{\pi_q(n)}], \quad (37)$$

where the variable τ satisfies

$$\boldsymbol{e}_{\pi_q(\tau)} = \boldsymbol{e}_r. \quad (38)$$

This result tells us that each component of the gradient of $f \in \mathcal{X}_T$ is a piecewise-constant function. Over the simplices whose orientation follows π_q, one can obtain the value of the rth component of the gradient by convolving the expansion coefficients c and the finite-difference filter $g_{\pi_q,r}$. To further clarify, we illustrate the filters $g_{\pi_q,r}$ in the two-dimensional case in Fig. 3. Each filter $g_{\pi_q,r}$ has only two nonzero values ($+1$ and (-1)) regardless of the dimensionality. The number of filters corresponding to each simplex orientation is d and the number of orientations is $d!$.

Proof of Lemma 1: We combine Property iv of the search space and (4) to write

$$\nabla f(\cdot) = \sum_{\boldsymbol{k} \in \mathbb{Z}^d} \sum_{q=1}^{d!} \boldsymbol{a}_{\pi_q,\boldsymbol{k},T} \mathbb{1}_{\pi_q,\boldsymbol{k},T}(\cdot). \quad (39)$$

To find $\boldsymbol{a}_{\pi_q,\boldsymbol{k},T}$, we use the hyperplane equation over the simplex $S_{\pi_q,\boldsymbol{k},T}$. The result is

$$T\mathbf{P}\boldsymbol{a}_{\pi_q,\boldsymbol{k},T} = \begin{bmatrix} c[\boldsymbol{k} + \boldsymbol{e}_{\pi_q(1)}] - c[\boldsymbol{k}] \\ \vdots \\ c[\boldsymbol{k} + \sum_{n=1}^{d} \boldsymbol{e}_{\pi_q(n)}] - c[\boldsymbol{k} + \sum_{n=1}^{d-1} \boldsymbol{e}_{\pi_q(n)}] \end{bmatrix}, \quad (40)$$

where

$$\mathbf{P} = \begin{bmatrix} \boldsymbol{e}_{\pi_q(1)}^\top \\ \vdots \\ \boldsymbol{e}_{\pi_q(d)}^\top \end{bmatrix}. \quad (41)$$

Since $\mathbf{P}$ is a permutation matrix, we have that $\mathbf{P}^\top\mathbf{P} = \mathbf{I}$. Therefore $\mathbf{P}^{-1} = \mathbf{P}^\top$, and

$$\mathbf{a}_{\pi_q,\boldsymbol{k},T} = \frac{1}{T}\mathbf{P}^\top \begin{bmatrix} c[\boldsymbol{k} + \boldsymbol{e}_{\pi_q(1)}] - c[\boldsymbol{k}] \\ \vdots \\ c[\boldsymbol{k} + \sum_{n=1}^{d} \boldsymbol{e}_{\pi_q(n)}] - c[\boldsymbol{k} + \sum_{n=1}^{d-1} \boldsymbol{e}_{\pi_q(n)}] \end{bmatrix}. \quad (42)$$

Consequently,

$$\boldsymbol{a}_{\pi_q,\boldsymbol{k},T}[r] = \frac{1}{T}\left(c[\boldsymbol{k} + \sum_{n=1}^{\tau-1} \boldsymbol{e}_{\pi_q(n)} + \boldsymbol{e}_r] - c[\boldsymbol{k} + \sum_{n=1}^{\tau-1} \boldsymbol{e}_{\pi_q(n)}]\right), \quad (43)$$

where τ satisfies (38) and $r \in \{1, \ldots, d\}$. Equation (43) allows us to rewrite the gradient as

$$\nabla f(\cdot) = \frac{1}{T} \sum_{\boldsymbol{k} \in \mathbb{Z}^d} \sum_{q=1}^{d!} \sum_{r=1}^{d} \left(c[\boldsymbol{k} + \sum_{n=1}^{\tau-1} \boldsymbol{e}_{\pi_q(n)} + \boldsymbol{e}_r] - c[\boldsymbol{k} + \sum_{n=1}^{\tau-1} \boldsymbol{e}_{\pi_q(n)}]\right) \boldsymbol{e}_r \mathbb{1}_{\pi_q,\boldsymbol{k},T}(\cdot). \quad (44)$$

After consideration of (37) and the convolution property, the proof of Proposition 1 is complete. □

Theorem 2 (Isotropic Total Variation): The isotropic total variation of the function $f \in \mathcal{X}_T$ is

$$\mathrm{TV}(f) = \frac{T^{d-1}}{d!} \sum_{\boldsymbol{k} \in \mathbb{Z}^d} \sum_{q=1}^{d!} \left(\sum_{r=1}^{d} \left((c * g_{\pi_q,r})[\boldsymbol{k}]\right)^2\right)^{\frac{1}{2}}, \quad (45)$$

where the filter $g_{\pi_q,r}$ satisfies (37).

Theorem 2 hints that, in order to calculate the exact continuous-domain TV of $f \in \mathcal{X}_T$ the following steps are needed: (i) for every simplex orientation π_q, convolve the expansion coefficients c and the finite-difference filters $g_{\pi_q,r}$ for $r \in \{1, \cdots, d\}$ to obtain d values at point $\boldsymbol{k}$; (ii) compute the ℓ_2

Authorized licensed use limited to: EPFL LAUSANNE. Downloaded on June 25,2024 at 10:28:57 UTC from IEEE Xplore. Restrictions apply.

norm of that d values at each point; (iii) repeat the procedure for all the $d!$ orientations and add the results at each point; (iv) sum the obtained values over the whole domain and scale the resulting scalar value by $\frac{T^{d-1}}{d!}$. The procedure resembles the conventional TV calculation except for the aggregation across the simplex orientations. For the conventional pixel-based TV, only one arbitrary orientation is considered.

Proof of Theorem 2: We first calculate the ℓ_2 norm of the gradient as

$$\|\boldsymbol{\nabla} f\|_2(\cdot) = \frac{1}{T}\sum_{\boldsymbol{k}\in\mathbb{Z}^d}\sum_{q=1}^{d!}\left(\sum_{r=1}^{d}((c*g_{\pi_q,r})[\boldsymbol{k}])^2\right)^{\frac{1}{2}}\mathbb{1}_{\pi_q,\boldsymbol{k},T}(\cdot). \tag{46}$$

Then, we obtain that

$$\begin{aligned}\mathrm{TV}(f) &= \frac{1}{T}\sum_{\boldsymbol{k}\in\mathbb{Z}^d}\sum_{q=1}^{d!}\left(\sum_{r=1}^{d}((c*g_{\pi_q,r})[\boldsymbol{k}])^2\right)^{\frac{1}{2}}\\ &\quad\int_{\mathbb{R}^d}\mathbb{1}_{\pi_q,\boldsymbol{k},T}(\boldsymbol{x})\mathrm{d}\boldsymbol{x}\\ &= \frac{T^{d-1}}{d!}\sum_{\boldsymbol{k}\in\mathbb{Z}^d}\sum_{q=1}^{d!}\left(\sum_{r=1}^{d}((c*g_{\pi_q,r})[\boldsymbol{k}])^2\right)^{\frac{1}{2}},\end{aligned} \tag{47}$$

which completes the proof. Note that the gradient of f on the junction of the domain simplices is not defined. However, the set of such junctions is of measure zero and does not affect the obtained result.

2) Hessian Total Variation: We first calculate the Hessian of $f\in\mathcal{X}_T$ in Lemma 2; we then compute its HTV in Theorem 3.

Lemma 2 (Explicit Hessian): Let the function $\psi_{p,q}:\mathbb{R}^d\to\mathbb{R}$ be defined as

$$\psi_{p,q}(\boldsymbol{x}) = \prod_{r\in\{1,\dots,d\}\setminus\{p,q\}}\mathrm{box}(x_r - x_p), \tag{48}$$

and let the functions $\mu_{1,p}$ and $\mu_{2,p,q}$ be defined as

$$\begin{aligned}\mu_{1,p}(\boldsymbol{x}) &= \delta(x_p)\psi_{p,p}(\boldsymbol{x})\\ \mu_{2,p,q}(\boldsymbol{x}) &= \delta(x_p - x_q)\mathrm{box}(x_p)\psi_{p,q}(\boldsymbol{x}),\end{aligned} \tag{49}$$

where $\boldsymbol{x} = (x_r)_{r=1}^d$. Further, we define the matrix $\mathbf{M}_{p,q}\in\mathbb{R}^{d\times d}$ such that it is zero everywhere except for its $[p,q]$-th component, which is equal to one. We define matrix $\mathbf{C}_{p,q}$ as

$$\mathbf{C}_{p,q} = \mathbf{M}_{p,q} + \mathbf{M}_{q,p} - \mathbf{M}_{p,p} - \mathbf{M}_{q,q}. \tag{50}$$

Then, the generalized Hessian of $f\in\mathcal{X}_T$ is

$$\begin{aligned}\mathbf{H}_f(\cdot) = \frac{1}{T^2}\Bigg(&\sum_{\boldsymbol{k}\in\mathbb{Z}^d}\sum_{p=1}^{d}(c*\kappa_{1,p})[\boldsymbol{k}]\mu_{1,p}\left(\frac{\cdot}{T}-\boldsymbol{k}\right)\mathbf{M}_{p,p}\\ &+\sum_{\boldsymbol{k}\in\mathbb{Z}^d}\sum_{p=1}^{d}\sum_{q=1}^{p}(c*\kappa_{2,p,q})[\boldsymbol{k}]\mu_{2,p,q}\left(\frac{\cdot}{T}-\boldsymbol{k}\right)\mathbf{C}_{p,q}\Bigg),\end{aligned} \tag{51}$$

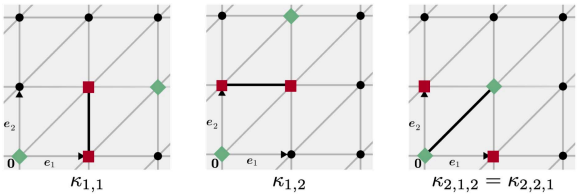

Fig. 4. Discrete filters that are used for the calculation of the generalized Hessian and HTV in $d=2$. The filters are zero everywhere except at the diamond and square points that correspond to the values of $+1$ and (-1), respectively. From left to right and at each plot, the black line depicts the facet over which the difference of gradients contributes to the HTV and is also the support of $\mu_{1,1}(\cdot-\boldsymbol{e}_1)$, $\mu_{1,2}(\cdot-\boldsymbol{e}_2)$, and $\mu_{2,1,1}$ respectively.

where

$$\begin{aligned}\kappa_{1,p}[\cdot] &= \delta[\cdot] - \delta[\cdot-\boldsymbol{e}_p] - \delta[\cdot+\mathbf{1}] + \delta[\cdot+\mathbf{1}-\boldsymbol{e}_p],\\ \kappa_{2,p,q}[\cdot] &= \delta[\cdot] - \delta[\cdot-\boldsymbol{e}_p] - \delta[\cdot-\boldsymbol{e}_q] + \delta[\cdot-\boldsymbol{e}_p-\boldsymbol{e}_q].\end{aligned} \tag{52}$$

In Lemma 2, we see that the Hessian of the CPWL function $f\in\mathcal{X}_T$ is zero everywhere except for the boundaries (facets) of the simplices of the Kuhn-Freudental triangulation. These boundaries are fully identified by the supports of the Dirac fences $\mu_{1,p}(\frac{\cdot}{T}-\boldsymbol{k})$ and $\mu_{2,p,q}(\frac{\cdot}{T}-\boldsymbol{k})$ for $\boldsymbol{k}\in\mathbb{Z}^d$ (see Fig. 4). The Hessian at these boundaries is equal to the matrices $\mathbf{M}_{p,p}$ or $\mathbf{C}_{p,q}$ multiplied by a scalar value. To obtain the proper scalar value, one needs to convolve the expansion coefficients c and the filters $\kappa_{1,p}$ or $\kappa_{2,p,q}$. We show these filters in the two-dimensional case in Fig. 4.

Proof of Lemma 2: By linearity and the chain rule for differentiation, we obtain that

$$\mathbf{H}_f(\cdot) = \frac{1}{T^2}\sum_{\boldsymbol{k}\in\mathbb{Z}^d}c[\boldsymbol{k}]\mathbf{H}_f\{\varphi\}\left(\frac{\cdot}{T}-\boldsymbol{k}\right)(\boldsymbol{x}). \tag{53}$$

We compute $\mathbf{H}_f\{\varphi\}$ (the Hessian of the basis function) in Appendix C. It follows

$$\begin{aligned}\mathbf{H}_f(\cdot) = \frac{1}{T^2}\sum_{\boldsymbol{k}\in\mathbb{Z}^d}\sum_{p=1}^{d}c[\boldsymbol{k}]\Bigg(&\mu_{1,p}\left(\frac{\cdot}{T}-\boldsymbol{k}+\mathbf{1}\right)\\ &+\mu_{1,p}\left(\frac{\cdot}{T}-\boldsymbol{k}-\boldsymbol{e}_p\right)-\mu_{1,p}\left(\frac{\cdot}{T}-\boldsymbol{k}-\boldsymbol{e}_p+\mathbf{1}\right)\\ &-\mu_{1,p}\left(\frac{\cdot}{T}-\boldsymbol{k}\right)\Bigg)\mathbf{M}_{p,p}\\ -\frac{1}{T^2}\sum_{\boldsymbol{k}\in\mathbb{Z}^d}\sum_{p=1}^{d}\sum_{q=1}^{q<p}\Bigg(&\mu_{2,p,q}\left(\frac{\cdot}{T}-\boldsymbol{k}+\mathbf{1}-\boldsymbol{e}_q\right)\\ &+\mu_{2,p,q}\left(\frac{\cdot}{T}-\boldsymbol{k}+\mathbf{1}-\boldsymbol{e}_p\right)-\mu_{2,p,q}\left(\frac{\cdot}{T}-\boldsymbol{k}+\mathbf{1}\right)\\ &-\mu_{2,p,q}\left(\frac{\cdot}{T}-\boldsymbol{k}+\mathbf{1}-\boldsymbol{e}_p-\boldsymbol{e}_q\right)\Bigg)\mathbf{C}_{p,q}.\end{aligned} \tag{54}$$

Authorized licensed use limited to: EPFL LAUSANNE. Downloaded on June 25,2024 at 10:28:57 UTC from IEEE Xplore. Restrictions apply.

By a change of variable, by the definition of the filters in (52), and by the property of convolution, (54) leads to (51), thereby completing the proof.

Theorem 3 (Hessian Total Variation): The Hessian total variation of the function $f \in \mathcal{X}_T$ is

$$\mathrm{HTV}(f) = T^{d-2} \left(\sum_{\boldsymbol{k} \in \mathbb{Z}^d} \sum_{p=1}^{d} |(c * \kappa_{1,p})[\boldsymbol{k}]| + 2 \sum_{\boldsymbol{k} \in \mathbb{Z}^d} \sum_{p=1}^{d} \sum_{q=1}^{p} |(c * \kappa_{2,p,q})[\boldsymbol{k}]| \right). \tag{55}$$

This shows that one can compute the continuous-domain HTV of $f \in \mathcal{X}_T$ by aggregating the results of the convolution of c with $\kappa_{1,p}$ and $\kappa_{2,p,q}$. Remarkably, this eliminates the need for the SVD that is otherwise required for the traditional computation of the Schatten norm. Each filter only has four nonzero values in any dimension. The number of the filters (convolutions) in dimension d is $\binom{d+1}{2}$.

Proof of Theorem 3: From (11), and noting that the individual arguments of the summations in (51) are disjoint Dirac fences, we obtain that

$$\mathrm{HTV}(f) = \tag{56}$$

$$\frac{1}{T^2} \left(\sum_{\boldsymbol{k} \in \mathbb{Z}^d} \sum_{p=1}^{d} \left\| (c * \kappa_{1,p})[\boldsymbol{k}] \mu_{1,p} \left(\frac{\cdot}{T} - \boldsymbol{k} \right) \mathbf{M}_{p,p} \right\|_{\mathcal{S}_1, \mathcal{M}} + \sum_{\boldsymbol{k} \in \mathbb{Z}^d} \sum_{p=1}^{d} \sum_{q=1}^{p} \left\| (c * \kappa_{2,p,q})[\boldsymbol{k}] \mu_{2,p,q} \left(\frac{\cdot}{T} - \boldsymbol{k} \right) \mathbf{C}_{p,q} \right\|_{\mathcal{S}_1, \mathcal{M}} \right). \tag{57}$$

We then invoke formula for the $\| \cdot \|_{\mathcal{S}_1, \mathcal{M}}$ a Dirac fence (see [26], Theorem 3.5-2), and obtain

$$\mathrm{HTV}(f) = \frac{1}{T^2} \sum_{\boldsymbol{k} \in \mathbb{Z}^d} \sum_{p=1}^{d} T \, |(c * \kappa_{1,p})[\boldsymbol{k}]| \, \mathrm{Leb}(D) \, \|\mathbf{M}_{p,p}\|_{\mathcal{S}_1} + \frac{1}{T^2} \sum_{\boldsymbol{k} \in \mathbb{Z}^d} \sum_{p=1}^{d} \sum_{q=1}^{p} T \, |(c * \kappa_{2,p,q})[\boldsymbol{k}]| \, \mathrm{Leb}(E) \, \|\mathbf{C}_{p,q}\|_{\mathcal{S}_1}, \tag{58}$$

where

$$D = \{ \boldsymbol{x} \in \mathbb{R}^{d-1} : \mathbf{box} \left(\frac{\boldsymbol{x}}{T} \right) = 1 \}$$

$$E = \{ (x_1, \boldsymbol{x}) \in \mathbb{R}^{d-1} : \mathrm{box} \left(\frac{x_1}{T} \right) \mathbf{box} \left(\frac{\boldsymbol{x} - x_1 \mathbf{1}}{T} \right) = 1 \}, \tag{59}$$

and Leb returns the Lebesgue measure (volume) of the input domain. The only nonzero eigenvalues of $\mathbf{M}_{p,p}$ and $\mathbf{C}_{p,q}$ are the elements of $\{1\}$ and $\{1, -1\}$, respectively. Therefore, we have that $\|\mathbf{M}_{p,p}\|_{\mathcal{S}_1} = 1$ and $\|\mathbf{C}_{p,q}\|_{\mathcal{S}_1} = 2$. In addition, $\mathrm{Leb}(D) = T^{d-1}$ and $\mathrm{Leb}(E) = T^{d-1}$. Hence, we obtain that

$$\mathrm{HTV}(f) = T^{d-2} \left(\sum_{\boldsymbol{k} \in \mathbb{Z}^d} \sum_{p=1}^{d} |(c * \kappa_{1,p})[\boldsymbol{k}]| + 2 \sum_{\boldsymbol{k} \in \mathbb{Z}^d} \sum_{p=1}^{d} \sum_{q=1}^{p} |(c * \kappa_{2,p,q})[\boldsymbol{k}]| \right), \tag{60}$$

which completes the proof. □

IV. Discretization

In this section, we discretize the continuous-domain problem using our multiscale CPWL search spaces. The representation of the signal f at scale s is denoted by $f_s \in \mathcal{X}_{(s)}$ and satisfies (32). Our model allows us to uniquely characterize the function $f_s : \mathbb{R}^d \to \mathbb{R}$ with the discrete expansion coefficients $\{c_s[\boldsymbol{k}]\}_{\boldsymbol{k} \in \mathbb{Z}^d}$.

Lemma 3: Let $\Omega = [0, b_1] \times \cdots \times [0, b_d] \subset \mathbb{R}^d$ and define $N_s = \lceil 2^s (\max_r b_r) \rceil + 1$ and

$$\mathcal{K}_s = \{0, \ldots, N_s - 1\} \times \ldots \times \{0, \cdots, N_s - 1\}. \tag{61}$$

Then,

$$\Omega \subseteq \mathrm{Hull}(\{2^{-s} \boldsymbol{k} | \boldsymbol{k} \in \mathcal{K}_s\}), \tag{62}$$

where Hull returns the convex hull of a set of input points.

We assume that the signal f to be recovered satisfies the zero boundary conditions

$$f(\boldsymbol{x}) = 0 \quad \text{for} \quad \boldsymbol{x} \notin \Omega \quad \text{or} \quad \boldsymbol{x} \in \partial\Omega, \tag{63}$$

where $\Omega = [0, b_1] \times \cdots \times [0, b_d] \subset \mathbb{R}^d$ and $\partial\Omega$ denotes the boundary of Ω. With this assumption and from Lemma 3, f_s is (exactly) equal to

$$f_s(\cdot) = \sum_{\boldsymbol{k} \in \mathcal{K}_s} c_s[\boldsymbol{k}] \varphi \left(2^s \cdot - \boldsymbol{k} \right), \tag{64}$$

where $\mathcal{K}_s$ satisfies (61) and $|\mathcal{K}_s| = N_s^d$. Since $\mathcal{K}_s$ is a finite set, one can define a bijective mapping $\mathrm{idx}_s : \mathcal{K}_s \to \mathbb{N}$, with preimage $\mathrm{idx}_s^{-1} : \mathbb{N} \to \mathcal{K}_s$, to rewrite (31) as

$$f_s(\cdot) = \sum_{n=1}^{N_s^d} c_{s,n} \varphi_{s,n}, \tag{65}$$

where $c_{s,n} = c_s(\mathrm{idx}_s^{-1}[n])$ and $\varphi_{s,n} = \varphi(2^s \cdot - \mathrm{idx}_s^{-1}[n])$. In this way, one obtains that $\boldsymbol{c}_s = \mathrm{Vec}(c_s) = (c_{s,n})_{n=1}^{N_s^d} \in \mathbb{R}^{N_s^d}$. This is the vectorized representation of signal at resolution s, which we refer to as the parameters of our model. As a result of Theorem 1, the relation between the parameters at successive scales is

$$\boldsymbol{c}_{s+1} = \mathbf{U}_s \boldsymbol{c}_s, \tag{66}$$

where $\mathbf{U}_s \in \mathbb{R}^{N_{s+1}^d \times N_s^d}$ corresponds to the transpose of a Toeplitz-like matrix associated with the convolution (with a stride of two) of c_s with the mirrored version of u defined in (29). More details are provided in Appendix D. Lemma 3, also verifies that $N_{s+1} = (2N_s - 1)$.

Authorized licensed use limited to: EPFL LAUSANNE. Downloaded on June 25,2024 at 10:28:57 UTC from IEEE Xplore. Restrictions apply.

A. Exact Discretization of the Forward Operator

Here, we describe how to discretize the forward operator using the parameters $\boldsymbol{c}_s$. For a linear forward operator $\boldsymbol{\nu}$ and at scale s, we have that

$$v_m(f_s) = \langle \nu_m, f_s \rangle = \left\langle \nu_m, \sum_{n=1}^{N_s^d} c_{s,n} \varphi_{s,n} \right\rangle = \sum_{n=1}^{N_s^d} c_{s,n} \langle \nu_m, \varphi_{s,n} \rangle = \boldsymbol{h}_{s,m}^\top \boldsymbol{c}_s, \tag{67}$$

where $m \in \{1, \dots, M\}$, and $\boldsymbol{h}_{s,m} = (\langle \nu_m, \varphi_{s,n} \rangle)_{n=1}^{N_s^d}$. By constructing $\mathbf{H}_s \in \mathbb{R}^{M \times N_s^d}$ as

$$\mathbf{H}_s = \begin{bmatrix} \boldsymbol{h}_{s,1}^\top \\ \vdots \\ \boldsymbol{h}_{s,M}^\top \end{bmatrix}, \tag{68}$$

we have that

$$\boldsymbol{v}(f_s) = \mathbf{H}_s \boldsymbol{c}_s. \tag{69}$$

The matrix $\mathbf{H}_s$ is the exact discretization of the forward operator at scale s and depends on the response of the box splines to the system. As a consequence of Theorem 1, we have that

$$\mathbf{H}_s = \mathbf{H}_{s+1} \mathbf{U}_s. \tag{70}$$

Therefore, we only need to compute the discretization of the forward operator at some fine scale s_{fine} and use (70) to propagate it to coarser scales. The computation of $\mathbf{H}_s$ can be further simplified depending on the properties of the forward operator.

1) Example: If the forward operator corresponds to a sampling in the Fourier domain, we can write that

$$\langle \nu_m, \varphi_{s,n} \rangle = \langle \mathrm{e}^{-\mathrm{j}\boldsymbol{\omega}_m^\top \cdot}, \varphi_{s,n} \rangle = \hat{\varphi}_{s,n}(\boldsymbol{\omega}_m) = 2^{-sd} \hat{\varphi}_{0,n}(2^{-s}\boldsymbol{\omega}_m) \tag{71}$$

$$= 2^{-sd} \hat{\varphi}(2^{-s}\boldsymbol{\omega}_m) \mathrm{e}^{-\mathrm{j}2^{-s}\boldsymbol{\omega}_m^\top \mathrm{idx}_s^{-1}[n]}. \tag{72}$$

The term $\hat{\varphi}(2^{-s}\boldsymbol{\omega}_m)$ can be computed in accordance with (18). If one then defines $\mathbf{w}_{s,m} = (\mathrm{e}^{-\mathrm{j}2^{-s}\boldsymbol{\omega}_m^\top \mathrm{idx}_s^{-1}[n]})_{n=1}^{N_s^d}$, it follows that

$$\boldsymbol{h}_{s,m} = 2^{-sd} \hat{\varphi}(2^{-s}\boldsymbol{w}_m) \mathbf{w}_{s,m} \tag{73}$$

and, therefore, that

$$\mathbf{H}_s = \mathbf{diag}(\hat{\boldsymbol{\varphi}}_s) \mathbf{W}_s, \tag{74}$$

where $\mathbf{diag} : \mathbb{R}^M \to \mathbb{R}^{M \times M}$ builds a diagonal matrix with the elements of the diagonal being the input vector $\hat{\boldsymbol{\varphi}}_s = (2^{-sd} \hat{\varphi}(2^{-s}\boldsymbol{\omega}_m))_{m=1}^M \in \mathbb{R}^M$, and

$$\mathbf{W}_s = \begin{bmatrix} \mathbf{w}_{s,1}^\top \\ \vdots \\ \mathbf{w}_{s,M}^\top \end{bmatrix} \in M \times \mathbb{C}^{N_s^d}. \tag{75}$$

By choosing $\boldsymbol{w}_m$ on a proper grid, one can use the d-dimensional discrete Fourier transform (DFT) followed by some vectorization to evaluate $\mathbf{W}_s \boldsymbol{c}_s$.

B. Exact Discretization of TV and HTV

Here, we present the formulas for TV and HTV of $f_s \in \mathcal{X}_{(s)}$ in terms of the parameters $\boldsymbol{c}_s$.

1) Total Variation: Through Theorem 2 and considering the vectorization of the expansion coefficients $\boldsymbol{c}_s = \mathrm{Vec}(c_s)$, we obtain that

$$\mathrm{TV}(f_s) = \frac{2^{-s(d-1)}}{d!} \left\| \mathbf{L}_{s,\mathrm{TV}} \otimes \boldsymbol{c}_s \right\|_{2,1}, \tag{76}$$

where $\mathbf{L}_{s,\mathrm{TV}} \in \mathbb{R}^{d! N_s^d \times d N_s^d}$ is defined as

$$\mathbf{L}_{s,\mathrm{TV}} = \begin{bmatrix} \mathbf{L}_{s,\pi_1,1} & \cdots & \mathbf{L}_{s,\pi_1,d} \\ \vdots & \ddots & \vdots \\ \mathbf{L}_{s,\pi_{d!},1} & \cdots & \mathbf{L}_{s,\pi_{d!},d} \end{bmatrix}. \tag{77}$$

Here $\mathbf{L}_{s,\pi_q,r} \in \mathbb{R}^{N_s^d \times N_s^d}$ is a Toeplitz-like matrix associated to the vectorization of the convolution of the expansioncoefficients c_s with the kernel $g_{\pi_q,r}$ defined in (37). We present the definition of the linear operator $(\mathbf{L} \otimes \cdot)$ and its adjoint in Appendix E. The $\|\cdot\|_{2,1}$ is defined as the application of the ℓ_2 norm to the rows of the input matrix and then that of the ℓ_1 norm to the resulting vector.

Our result for the TV in (76) resembles the conventional pixel-based discretization, except that the present finite-difference filters are tailored to our model, which makes our computations exact.

2) Hessian Total Variation: Through Theorem 3, we have that

$$\mathrm{HTV}(f_s) = 2^{-s(d-2)} \left\| \mathbf{L}_{s,\mathrm{HTV}} \boldsymbol{c}_s \right\|_1. \tag{78}$$

The matrix $\mathbf{L}_{s,\mathrm{HTV}} \in \mathbb{R}^{\frac{d(d+1)}{2} N_s^d \times N_s^d}$ is defined as

$$\mathbf{L}_{s,\mathrm{HTV}} = \begin{bmatrix} \mathbf{L}_{s,1} \\ \vdots \\ \mathbf{L}_{s,d} \\ 2\mathbf{L}_{s,2,1} \\ \vdots \\ 2\mathbf{L}_{s,d,d-1} \end{bmatrix}, \tag{79}$$

where $\mathbf{L}_{s,p} \in \mathbb{R}^{N_s^d \times N_s^d}$ is a Toeplitz-like matrix that corresponds to the vectorization of the convolution of c_s with the filter $\kappa_{1,p}$ for $p \in \{1, \dots, d\}$, and $\mathbf{L}_{s,p,q} \in \mathbb{R}^{N_s^d \times N_s^d}$ is a Toeplitz-like matrix that corresponds to the vectorization of the convolution of c_s with the filter $\kappa_{2,p,q}$ of (52) for $p, q \in \{1, \dots, d\}, q < p$.

The use of the Hessian total-variation regularization, defined with different Schatten-p norms, was pioneered by Lefkimmiatis et al. [28]. In their implementation, they estimate the Hessian using second-order finite differences and rely on a Riemann-sum approximation of the integral for the computation of the seminorm. For $p = 2$, the Schatten-2 norm coincides with the Frobenius norm and allows for a straightforward computation of the seminorm and its proximal operator. However, for $p = 1$, their computations rely on the SVD [29]. The present contribution is a refined scheme where the computation of HTV with

Authorized licensed use limited to: EPFL LAUSANNE. Downloaded on June 25,2024 at 10:28:57 UTC from IEEE Xplore. Restrictions apply.

Algorithm 1: Multiresolution Solver.

Initialize: $s = s_0 \in \mathbb{Z}, \epsilon_1 > 0$;
Initialize: $\boldsymbol{c}_{s_0} = \boldsymbol{0} \in \mathbb{R}^{N_{s_0}^d}, \boldsymbol{d}_{s_0} = \boldsymbol{0} \in \mathbb{R}^{N_{s_0}^d}$;
Define: $\text{Loss}_s(\boldsymbol{c}) = \frac{1}{2}\|\boldsymbol{y} - \text{H}_s\{\boldsymbol{c}\}\|_2^2 + \lambda \mathcal{R}_s(\boldsymbol{c})$;
$\hat{\boldsymbol{c}}_{s_0} = \textbf{Solver}_s(\boldsymbol{c}_{s_0}, \boldsymbol{d}_{s_0})$;
for $s \in \mathbb{Z}$ **and** $s > s_0$ **do**
  $\hat{\boldsymbol{c}}_s = \textbf{Solver}_s(\boldsymbol{c}_s, \boldsymbol{d}_s)$;
  $\boldsymbol{c}_{s+1} = \mathbf{U}_s \hat{\boldsymbol{c}}_s$;
  $\boldsymbol{d}_{s+1} = \mathbf{U}_s \hat{\boldsymbol{c}}_s$;
  if $(\text{Loss}_{s-1}(\hat{\boldsymbol{c}}_{s-1}) - \text{Loss}_s(\hat{\boldsymbol{c}}_s)) \leq \epsilon_1$ **then**
    return s, $\hat{f}_s = \sum_{\boldsymbol{k} \in \mathcal{K}_s} \hat{c}_s[\boldsymbol{k}] \varphi(2^s \cdot - \boldsymbol{k})$
  end if
end for

Algorithm 2: $\textbf{Solver}_s(\boldsymbol{c}_0, \boldsymbol{d}_0)$.

Initialize: $t_0 = 1, \alpha_1 = 1/\text{ct}_{\text{H}_s}, \gamma = \alpha_1 \lambda, \epsilon_2, \delta > 0$;
for $k = 0 \to N_{\text{iter}_1}$ **do**
  $\boldsymbol{w}_k = \boldsymbol{d}_k + \alpha_1 \text{Re}(\text{H}_{s,\text{adj}}\{\boldsymbol{y} - \text{H}_s\{\boldsymbol{d}_k\}\})$;
  $\epsilon_{3,k} = \min(\delta, \frac{1}{(k+1)^3})$;
  $\boldsymbol{c}_{k+1} = \textbf{Prox}_{\gamma, \mathcal{R}_s}(\boldsymbol{w}_k, \epsilon_{3,k})$;
  $t_{k+1} = \frac{1+\sqrt{4t_k^2+1}}{2}$;
  $\boldsymbol{d}_{k+1} = \boldsymbol{c}_{k+1} + \frac{t_k - 1}{t_{k+1}}(\boldsymbol{c}_{k+1} - \boldsymbol{c}_k)$;
  $e_k = |\text{Loss}_s(\boldsymbol{c}_k) - \text{Loss}_s(\boldsymbol{c}_{k+1})|$;
  if $e_k < \epsilon_2 \text{Loss}_s(\boldsymbol{c}_k)$ **then**
    break
  end if
end for
return $\boldsymbol{c}_{k+1}$

Schatten-1 norm is not only exact but also does not require SVD, which speeds up the computations.

V. Optimization

By choosing the mean-squared error as our loss functional and by using our developed search spaces $\{\mathcal{X}_{(s)}\}$, we obtain a solution

$$\hat{f}_s \in \arg\min_{f_s \in \mathcal{X}_{(s)}} J(f_s) \tag{80}$$

At scale $s \in \mathbb{Z}$, where $J(f_s) = (\|\boldsymbol{\nu}(f_s) - \boldsymbol{y}\|_2^2 + \lambda \mathcal{R}(f_s))$. Because of the embedding of the search spaces, we necessarily have that $J(\hat{f}_{s+1}) \leq J(\hat{f}_s)$. Our aim is then to find the scale s_{fine} for which $J(\hat{f}_{s_{\text{fine}}}) \approx J(\hat{f}_{s_{\text{fine}}+1})$. In other words, s_{fine} is a scale beyond which the refinement of the search space no longer improves the optimization cost.

We present our multiresolution scheme in Algorithm 1. At each scale s, we discretize (80) exactly using (69), (76), and (78) as

$$\hat{\boldsymbol{c}}_s \in \arg\min_{\boldsymbol{c} \in \mathcal{C}_s} \text{Loss}_s(\boldsymbol{c}), \tag{81}$$

with $\text{Loss}_s(\boldsymbol{c}) = (\frac{1}{2}\|\boldsymbol{y} - \text{H}_s\{\boldsymbol{c}\}\|_2^2 + \lambda \mathcal{R}_s(\boldsymbol{c}))$. There, $\mathcal{C}_s \subseteq \mathbb{R}^{N_s^d}$, while $\text{H}_s : \mathbb{R}^{N_s^d} \to \mathbb{C}^M$ is a linear operator such that $\text{H}_s\{\boldsymbol{c}\} = \mathbf{H}_s \boldsymbol{c}$ with adjoint $\text{H}_{s,\text{adj}}\{\boldsymbol{v}\} = \mathbf{H}_s^\top \boldsymbol{v}$. Moreover, $\mathcal{C}_s$ encodes additional conditions on the range of the function $f_s \in \mathcal{X}_{(s)}$. In the absence of such additional constraints, we have that $\mathcal{C}_s = \mathbb{R}^{N_s^d}$. For our model, due to the interpolatory property of the basis, the non-negativity of the function f_s implies the non-negativity of its coefficients $\boldsymbol{c}_s$. Under that constraint, we have that $\mathcal{C}_s = \mathbb{R}_{\geq 0}^{N_s^d}$. For the TV and HTV regularizers, $\mathcal{R}_s(\boldsymbol{c})$ is of the form

$$\mathcal{R}_s(\boldsymbol{c}) = \|\text{L}_s\{\boldsymbol{c}\}\|, \tag{82}$$

where $\|\cdot\| : \mathbb{R}^{N_{L_s}} \to \mathbb{R}^+$ denotes an appropriate discrete (mixed) norm, and $\text{L}_s : \mathbb{R}^{N_s} \to \mathbb{R}^{N_{L_s}}$ is a linear operator. We present more details in Table I.

Algorithm 3: Computation of $\textbf{Prox}_{\gamma, \mathcal{R}_s}(\boldsymbol{z}, \epsilon_3)$.

Initialize: $\boldsymbol{u}_0 = \boldsymbol{0} \in \mathbb{R}^{N_{L_s}}, \boldsymbol{v}_0 = \boldsymbol{0} \in \mathbb{R}^{N_{L_s}}, t_0 = 1$;
Initialize: $\alpha_2 = 1/(\text{ct}_{\mathcal{R}_s} \gamma)$;
Define: $\text{Pr}_s(\boldsymbol{c}) = \frac{1}{2}\|\boldsymbol{z} - \boldsymbol{c}\|_2^2 + \lambda \mathcal{R}_s(\boldsymbol{c})$;
Define: $\text{Du}_s(\boldsymbol{c}) = \frac{1}{2}(\|\boldsymbol{c} - \textbf{proj}_{\mathcal{C}_s}(\boldsymbol{c})\|_2^2 - \|\boldsymbol{c}\|_2^2 + \|\boldsymbol{z}\|_2^2)$;
for $k = 0 \to N_{\text{iter}_2}$ **do**
  $\boldsymbol{c}_k = \boldsymbol{z} - \gamma \text{L}_{s,\text{adj}}\{\boldsymbol{v}_k\}$;
  $\boldsymbol{c}_{\textbf{proj}_k} = \textbf{proj}_{\mathcal{C}_s}\{\boldsymbol{c}_k\}$;
  $\boldsymbol{u}_{k+1} = \mathbf{P}_{\mathcal{R}_s}(\boldsymbol{v}_k + \alpha_2 \text{L}_s\{\boldsymbol{c}_{\textbf{proj}_k}\})$;
  $t_{k+1} = \frac{1+\sqrt{4t_k^2+1}}{2}$;
  $\boldsymbol{v}_{k+1} = \boldsymbol{u}_{k+1} + \frac{t_k - 1}{t_{k+1}}(\boldsymbol{u}_{k+1} - \boldsymbol{u}_k)$;
  $\text{Gap}_k = \text{Pr}_s(\boldsymbol{c}_{\textbf{proj}_k}) - \text{Du}_s(\boldsymbol{c}_k)$;
  if $\text{Gap}_k < \epsilon_3$ **then**
    break
  end if
end for
return $\textbf{proj}_{\mathcal{C}_s}\{\boldsymbol{z} - \gamma \text{L}_{s,\text{adj}}\{\boldsymbol{u}_{k+1}\}\}$

TABLE I
Linear Operators and Norms Associated With Regularizers

Reg.	TV	HTV
$\|\cdot\|$	$\|\cdot\|_{2,1}$	$\|\cdot\|_1$
$\text{L}_s\{\boldsymbol{c}\}$	$\frac{2^{-s(d-1)}}{d!}\mathbf{L}_{s,\text{TV}} \otimes \boldsymbol{c}$	$2^{-s(d-2)}\mathbf{L}_{s,\text{HTV}}\boldsymbol{c}$
N_{L_s}	$d! N_s^d \times d$	$\frac{d(d+1)}{2} N_s^d \times 1$
$\text{L}_{s,\text{adj}}\{\boldsymbol{v}\}$	$\frac{2^{-s(d-1)}}{d!}\overline{\mathbf{L}}_{s,\text{TV}} \odot \boldsymbol{v}$	$2^{-s(d-2)}\mathbf{L}_{s,\text{HTV}}^\top \boldsymbol{v}$
$\mathbf{P}_{\mathcal{R}_s}$	**Normalize**	**Clip**

Once we have solved (81), we obtain the exact solution to (80) by invoking the fact that $\hat{f}_s = \sum_{\boldsymbol{k} \in \mathcal{K}_s} \hat{c}_s[\boldsymbol{k}] \varphi(2^s \cdot - \boldsymbol{k})$. We have that $J(\hat{f}_s) = \text{Loss}_s(\hat{\boldsymbol{c}}_s) = \text{Loss}_{s+1}(\mathbf{U}_s \hat{\boldsymbol{c}}_s)$. The latter equality guarantees the continuity of the loss with respect to a change of scale. Further, by optimization at scale $(s+1)$, we have that $J(\hat{f}_{s+1}) = \text{Loss}_{s+1}(\hat{\boldsymbol{c}}_{s+1}) \leq \text{Loss}_{s+1}(\mathbf{U}_s \hat{\boldsymbol{c}}_s) = J(\hat{f}_s)$. Therefore, the sequence of final losses at each scale is non-increasing.

We now develop iterative schemes to solve (81). We present our results in Algorithms 2 and 3. The derivation of the steps of these algorithms involves a dual formulation of the problem similar to [59] and [60]. Since Algorithm 2 requires the evaluation of a proximal operator without a closed form, we use an

Authorized licensed use limited to: EPFL LAUSANNE. Downloaded on June 25,2024 at 10:28:57 UTC from IEEE Xplore. Restrictions apply.

inexact-proximal method to accelerate the computations [61], [62], [63].

The last ingredients of our proposed algorithms are the functions $\mathbf{Clip} : \mathbb{R}^K \to \mathbb{R}^K$ and $\mathbf{Normalize} : \mathbb{R}^{K\times N} \to \mathbb{R}^{K\times N}$, for $k \in \{1, \ldots, K\}$ as

$$[\mathbf{Clip}(\boldsymbol{a})]_k = \frac{a_k}{\max(|a_k|, 1)}, \tag{83}$$

with $\boldsymbol{a} = (a_k)_{k=1}^K$, and

$$[\mathbf{Normalize}(\mathbf{A})]_k = \frac{1}{\max(\|[\mathbf{A}]_k\|_2, 1)}[\mathbf{A}]_k, \tag{84}$$

where $[\mathbf{A}]_k$ denotes the kth row of the matrix $\mathbf{A} \in \mathbb{R}^{K\times N}$. Moreover, the constants $\mathrm{ct_H}$ and $\mathrm{ct}_\mathcal{R}$ are equal to the square of the spectral norm of the operators H and L, respectively. The function Re returns the complex input's real part, and $\|\cdot\|_F$ denotes the Frobenius norm.

Note that the Chambolle-Pock algorithm [64] could offer an alternative to Algorithm 2 for $\mathcal{C} = \mathbb{R}^{N_s^d}$. One might also consider methods based on primal-dual splitting [65], [66], [67] in the presence of constraints, with the caveat that they would involve additional hyperparameters.

A. Convergence to the Solution Over $\mathcal{B}_\mathcal{R}$

In the one-dimensional case, which is equivalent to our framework for $(d = 1)$, the authors in [37] show that the B-spline-based model can approximate the solution of the problem over $\mathcal{B}_{\mathrm{TV}}$ and $\mathcal{B}_{\mathrm{HTV}}$.

For $d \in \{2, 3\}$, the proposed CPWL functions can approximate any integrable function and its TV [48], [68], [69]. This approximation result serves as a strong indication that our proposed framework could approximate the solution of (2) over $\mathcal{B}_{\mathrm{TV}}$. Moreover, CPWL functions can approximate any locally integrable function and its HTV [32]. The construction of those CPWL functions depends on adaptive meshes. While our framework allows for the exact solution of (2) over a search space of CPWL functions on uniform grids, a proof of the convergence over $\mathcal{B}_{\mathrm{TV}}$ and $\mathcal{B}_{\mathrm{HTV}}$ as the grid size goes to zero would require further rigorous analyses and remains an open theoretical question.

VI. Experiments

In this section, we present our numerical experiments. Our code is available on the GitHub repository.[3] The reported times for CPU and GPU correspond to the execution of the code on an Intel(R) Xeon(R) Gold 6240 @ 2.60 GHz and on a Tesla V100-SXM2-32 GB, respectively.

A. Multiresolution Reconstruction

Here, we validate our multiresolution scheme to solve inverse problems in two setups: 1) a perfect reconstruction scenario with no regularization; and 2) a compressed-sensing counterpart with continuous-domain sparsity constraints. We solve all the problems under non-negativity constraints ($\mathcal{C}_s = \mathbb{R}^{N_s^d}_{\geq 0}$). To compute the peak-signal-to-noise ratio (PSNR) and to illustrate the continuous-domain signals, we consider samples on a fine (4096×4096) grid.

1) Perfect Reconstruction (PR): To test our approach, we intentionally express our continuous-domain ground truth using the box spline φ so that it is exactly representable in $\mathcal{X}_{(s)}$ for $s \geq 0$. It reads

$$f_{\mathrm{GT,PR}} = \sum_{k_1=0}^{512}\sum_{k_2=0}^{512} c_{\mathrm{GT}}[\boldsymbol{k}]\varphi(\cdot - \boldsymbol{k}), \tag{85}$$

where f_{GT} satisfies zero boundary conditions outside of the domain $[0, 512] \times [0, 512]$. Consequently, the expansion coefficients c_{GT} are zero on the boundaries. We choose a medical image from [70] as the basis for our ground truth.

We formulate an inverse problem in a regular imaging scenario where the perfect reconstruction of the ground truth is possible. To do so, we take advantage of (74) to calculate the Fourier transform of f_{GT} at the frequencies

$$\boldsymbol{\omega} \in \left\{-\pi, \ldots \frac{-\pi}{256}, 0, \frac{\pi}{256}, \ldots, \frac{255\pi}{256}\right\} \times \left\{-\pi, \ldots \frac{-\pi}{256}, 0, \frac{\pi}{256}, \ldots, \frac{255\pi}{256}\right\}. \tag{86}$$

To reconstruct the signal from the measurements, we use Algorithm 1 with $s_0 = (-3)$, and no regularization ($\lambda = 0$). Our conclusion from this experiment is that we can reconstruct the ground truth with high precision (PSNR > 100) as soon as we reach scale $s = 0$ with $J(\hat{f}_0) = 0$.

2) Compressed Sensing (CS): In this part, we focus on a compressed-sensing setup where the problem is ill-posed due to the small number of measurements. Our aims are twofold: to compare the effect of the TV versus HTV regularizations; and to compare our approach to the standard pixel-based approach to inverse problems, where an exact computation of TV is also possible (See in Appendix F how we embed pixel-based TV within a multiresolution scheme).

Here, we use a ground truth built upon a two-dimensional cubic-spline basis that is not representable by either the CPWL or the pixel bases. Specifically, we use

$$f_{\mathrm{GT,CS}} = \sum_{k_1=0}^{512}\sum_{k_2=0}^{512} c_{\mathrm{GT}}[\boldsymbol{k}]\psi(\cdot - \boldsymbol{k}) \tag{87}$$

with $\psi(\boldsymbol{x}) = \beta^3(x_1)\beta^3(x_2)$, where β^3 is a one-dimensional cubic spline [38].

To generate the measurements, we sample the continuous-domain Fourier transform of $f_{\mathrm{GT,CS}}$ at frequencies

$$\boldsymbol{\omega} \in \left\{-\frac{\pi}{2}, \ldots -\frac{\pi}{256}, 0, \frac{\pi}{256}, \ldots, \frac{127\pi}{256}\right\} \times \left\{-\frac{\pi}{2}, \ldots -\frac{\pi}{256}, 0, \frac{\pi}{256}, \ldots, \frac{127\pi}{256}\right\}, \tag{88}$$

which forms a grid of size (256×256). We then apply a radial mask and keep only 30 percent of the frequencies in (88) [71].

[3] https://github.com/mehrsapo/BoxDis/

Authorized licensed use limited to: EPFL LAUSANNE. Downloaded on June 25,2024 at 10:28:57 UTC from IEEE Xplore. Restrictions apply.

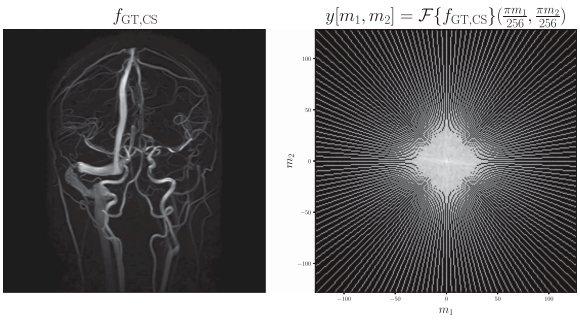

Fig. 5. Ground-truth image [70] and measurements for the CS setup.

Eventually, we retain only 7.5 percent of the frequencies of the perfect reconstruction scenario in (86). We illustrate the ground-truth signal and the measurements in Fig. 5.

To solve the reconstruction problems, we use Algorithm 1 with $s_0 = (-3)$, $\epsilon_1 = 10^{-4}$, $N_{\text{iter}_1} = 2000, N_{\text{iter}_2} = 500$, $\epsilon_2 = 10^{-6}$ and $\delta = 10^{-3}$. The regularization hyperparameter is $\lambda = 10^{-6}$ in all cases. For our experiments, Algorithm 1 stops at scale $s = 2$; therefore, it involves six scales $s \in \{-3, -2, -1, 0, 1, 2\}$. The corresponding grid sizes are $T \in \{8, 4, 2, 1, 0.5, 0.25\}$ and the number of grid points at each scale is $(N_{s_1} \times N_{s_2}) \in \{(65 \times 65), (129 \times 129), (257 \times 257), (513 \times 513), (1025 \times 1025), (2049 \times 2049)\}$.

In Fig. 6, we show the continuous-domain solutions ($\hat{f}_2$) of our three different approaches: (i) pixel-based TV; (ii) CPWL-based TV; and (iii) CPWL-based HTV. The PSNR values for the solutions are 35.03, 37.89, and 39.50, respectively. The TV-regularization-based solutions are formed of piecewise-constant pieces, which matches the results of the TV representer theorems [31]. Although the locations of the piecewise-constant regions are similar, the boundaries are rectangular for the pixel-based approach and less angular for the CPWL-based approach. By contrast, the HTV solution is locally piecewise-affine and. Hence, it can better reproduce the ground-truth structure.

In Fig. 7, we study the same solutions at different scales. At coarser scales, the loss is dominated by the data-fidelity term so that the regularization effect becomes more evident only at finer scales. We report the optimization loss in Fig. 8 for different approaches. We observe that the optimization loss is continuous and decreases with the change of scale as supported by theory. We also observe that the optimization cost stabilizes one a certain point of refinement has been reached.

To compare the computational efficiency of the various methods, we present the number of iterations of Algorithm 2 and the corresponding computational times to reach the stopping criteria at each scale in Tables II and III, respectively. Note that the relation between the two is not necessarily linear due to the inexact proximal scheme. We also report the number of iterations and time for the direct solution of the problem on the finest grid ($s = 2$) with zero initialization. The total time of our multiresolution scheme with the CPWL basis is less than the direct solution on the finest grid with zero initialization for both TV and HTV regularization. This computational acceleration is due to the better initialization of the problem by the solution at the previous scale.

TABLE II
NUMBER OF ITERATIONS OF ALGORITHM 2 WITH $\epsilon_2 = 10^{-6}$

Multiresolution s	Pixel, TV	CPWL, TV	CPWL, HTV
-3	27	38	38
-2	145	161	161
-1	1418	1536	1330
0	1738	1148	755
1	1826	862	481
2	1491	670	293
Direct $s = 2$	1831	1454	992

TABLE III
GPU TIME (SECONDS) OF ALGORITHM 2 WITH $\epsilon_2 = 10^{-6}$

Multiresolution s	Pixel, TV	CPWL, TV	CPWL, HTV
-3	0.72	0.86	0.84
-2	0.49	0.79	0.71
-1	10.85	10.61	15.42
0	81.41	46.49	60.07
1	948.86	261.68	320.67
2	3890.91	2169.57	874.37
Total	4738	2490.0	1272
Direct $s = 2$	4933	6460	3076

B. CPWL-Based Discrete Regularization

Our CPWL-based TV and HTV computations are essential for us to develop a multiresolution solver for inverse problems with a continuous-domain perspective. At a fixed scale, the resulting expressions (76) and (78) can be interpreted as discretization schemes for TV and HTV. One could therefore use these expressions as alternative definitions for the TV and HTV of discrete signals. In this section, we compare these discretizations to some of their classic counterparts.

1) Total Variation: For a discrete image $\boldsymbol{c} \in \mathbb{R}^{N_1 \times N_2}$, our CPWL-based computations for TV lead to

$$\text{TV}_{\text{CPWL}}(\boldsymbol{c}) = \frac{1}{2} \sum_{m=0}^{N_1-1} \sum_{n=0}^{N_2-1} \Big(\big((\boldsymbol{c}[m,n] - \boldsymbol{c}[m+1,n])^2 + (\boldsymbol{c}[m,n] - \boldsymbol{c}[m,n+1])^2\big)^{\frac{1}{2}} + \Big(\big(\boldsymbol{c}[m,n] - \boldsymbol{c}[m-1,n]^2 + (\boldsymbol{c}[m,n] - \boldsymbol{c}[m,n-1])^2\big)^{\frac{1}{2}}\Big), \tag{89}$$

with some proper zero padding. Expression (89) resembles the standard definition of discrete isotropic TV [46]. In effect, TV_{CPWL} is the result of averaging the standard discrete isotropic TV with backward and forward finite differences. This averaging results in TV_{CPWL} being invariant to rotations of 180 degrees, but increases the computational complexity of the optimization. To study this effect, we perform a denoising experiment with a ground-truth image of size (400×400). The ground-truth image involves two regions, a piecewise-constant disk and a smooth

Authorized licensed use limited to: EPFL LAUSANNE. Downloaded on June 25,2024 at 10:28:57 UTC from IEEE Xplore. Restrictions apply.

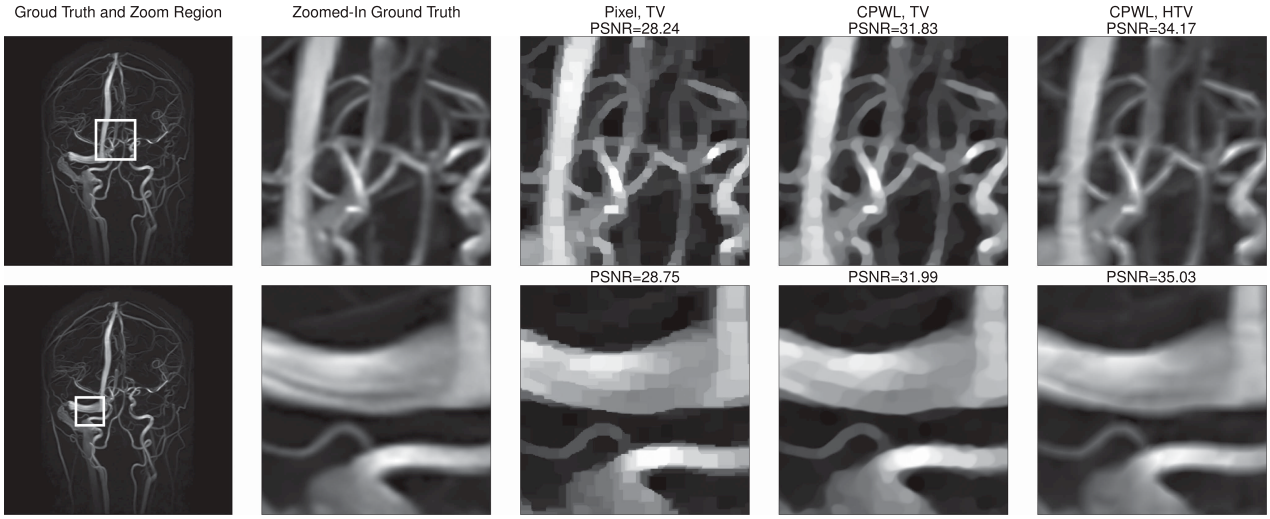

Fig. 6. Solutions of several multiresolution schemes after convergence at the finest scale.

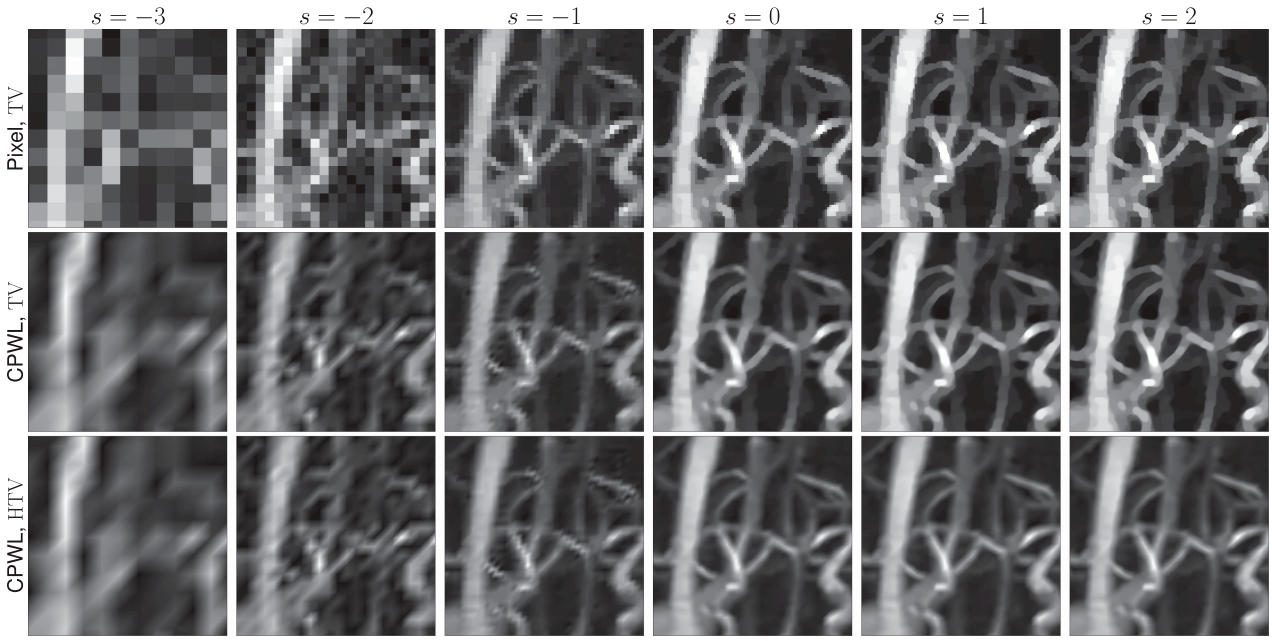

Fig. 7. Solutions at each scale. The region corresponds to the first row of Fig. 6.

region corresponding to the top view of a cone (Fig. 9(a)). The measurements are obtained by adding zero-mean Gaussian white noise with a standard variation of $50/255$. We compare our approach to the discrete isotropic TV with forward finite differences, and to the upwind TV—a discrete variant of TV that is invariant to ± 90 and 180 degrees rotations [72], [73]. To solve the optimization problem for all methods, we use a variant of Algorithm 3 with a stopping criterion of 10^{-6} on the relative error of the loss objective. We set the regularization hyperparameter to $\lambda = 0.5$ for all the methods. We present the PSNRs of the solutions (left half: disk, right half: cone), the number of iterations to reach the stopping criterion, and the computational times in Table IV.

TABLE IV
DISCRETE REGULARIZERS

Method	PSNR		Time (Seconds)		Iterations
	Disk	Cone	CPU	GPU	
TV Isotropic	34.18	36.79	9.26	0.10	165
TV Upwind	36.14	36.50	11.70	0.17	193
TV CWPL	34.26	36.88	19.15	0.16	172
HTV SVD-C	26.45	38.22	97.58	26.47	7319
HTV CPWL	25.17	39.36	3.63	0.87	259

The CPWL-based TV performs marginally better than the standard isotropic TV with a forward finite difference. This is

Authorized licensed use limited to: EPFL LAUSANNE. Downloaded on June 25,2024 at 10:28:57 UTC from IEEE Xplore. Restrictions apply.

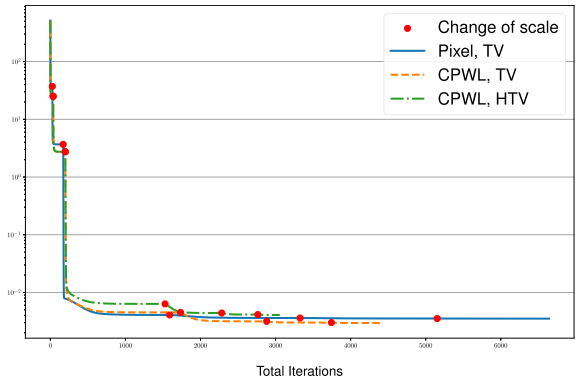

Fig. 8. Loss function for each method in terms of the total number of iterations of Algorithm 2 at different scales.

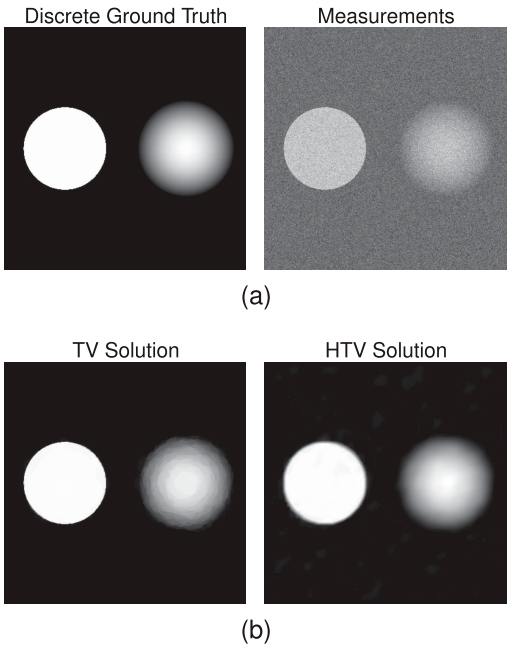

Fig. 9. (a) Discrete ground-truth signal and its noisy version as the measurements. (b) Solutions.

at the cost of CPU time. The upwind TV outperforms the other methods in the disk region but is not helpful in the cone region, for which HTV is much better suited.

2) Hessian Total Variation: We now repeat the denoising experiment of Section VI.B.1, but with HTV regularization (Fig. 9(b)). On the one hand, we observe that the HTV regularization reproduces the smooth structures much better than the TV-based solution, which shows staircase effects. On the other hand, the edge of the disk is blurred in the HTV solution and TV regularization performs better on such sharp edges.

We also compare our approach to the finite-difference-based approach for the Hessian-Schatten-1 regularization proposed in [29]. We present the results in Table IV. Our CPWL-based HTV regularization requires fewer iterations to reach the stopping criterion, accelerating the computations compared to the finite-difference-based approach. This is mainly because our exact computations for HTV lead to a minimization problem with ℓ_1-norm constraint, whereas that of [29] involves constraints with the Schatten-1 norm.

TABLE V
EVALUATION TIME FOR 200 ITERATIONS OF THE PROXIMAL OPERATOR OF HTV REGULARIZATION

	CPU Time			GPU Time		
d	SVD	SVD-C	CPWL	SVD	SVD-C	CPWL
2	101.68	4.18	4.73	19.70	0.79	0.14
3	9781.42	1258.90	434.51	2141.91	110.23	16.01

The faster convergence regarding the number of iterations comes with no compromise on the computational cost per iteration. On top of the Hessian approximation in [29], their iterative proximal operator requires a $(d \times d)$ SVD for each pixel and iteration. Our CPWL-based proximal operator for HTV requires the same number of convolutions but eliminates the need for SVDs (Algorithm 3). This means that we avoid a cost of $\mathcal{O}(d^3)$ per coefficient (pixel). To demonstrate this effect, we provide the execution times of 200 iterations of the evaluation of the HTV proximal operator for $d = 2, 3$ in Table V. There, for $d = 2$ and $d = 3$, we use an image and a volume with sizes of (583×493) pixels and $(583 \times 493 \times 40)$ voxels, respectively. For the proximal operator of [28], we follow two different approaches to calculate the SVD: the numerical PyTorch SVD function (SVD), and a closed-form solution (SVD-C). We note that implementing SVD-C for $d = 3$ was especially complicated.

Our method is faster than the finite-difference-based method in low dimensions. We expect the improvement to be even bigger in higher dimensions because there is no closed form for the SVD for $d > 4$.

VII. CONCLUSION

In this paper, we have provided a novel framework for the solution of the inverse problem (2) with continuous-domain total variation (TV) and Hessian total variation (HTV) regularization. We presented an exact discretization scheme based on box splines that takes into account the analog nature of the signals and the physical measurement set. Our formulation is exact numerically and yields discrete convolutional formulas for the continuously-defined TV and HTV. It also allows for straightforward evaluation of various forward operators in imaging. We also proposed a multiresolution scheme that speeds up computations and selects the optimal resolution for image reconstruction. In all our experiments, we reached a scale beyond which making the grid finer yields no further decrease in the optimization cost.

REFERENCES

[1] M. T. McCann and M. Unser, "Biomedical Image Reconstruction: From the Foundations to Deep Neural Networks," *Found. Trends Signal Process.*, vol. 13, 2019, pp. 283–359.
[2] P. Bertero, M. Boccacci, and C. De Mol, *Introduction to Inverse Problems in Imaging*, 2nd ed. Boca Raton, FL, USA, 2021.

Authorized licensed use limited to: EPFL LAUSANNE. Downloaded on June 25,2024 at 10:28:57 UTC from IEEE Xplore. Restrictions apply.

[3] G. Steidl, J. Weickert, T. Brox, P. Mrázek, and M. Welk, "On the equivalence of soft wavelet shrinkage, total variation diffusion, total variation regularization, and sides," *SIAM J. Numer. Anal.*, vol. 42, no. 2, pp. 686–713, 2004.

[4] D. L. Donoho, "Compressed sensing," *IEEE Trans. Inf. Theory*, vol. 52, no. 4, pp. 1289–1306, Apr. 2006.

[5] E. Candès and J. Romberg, "Sparsity and incoherence in compressive sampling," *Inverse Problems*, vol. 23, no. 3, 2007, Art. no. 969.

[6] M. A. T. Figueiredo, J. M. Bioucas-Dias, and R. D. Nowak, "Majorization–minimization algorithms for wavelet-based image restoration," *IEEE Trans. Image Process.*, vol. 16, no. 12, pp. 2980–2991, Dec. 2007.

[7] A. M. Bruckstein, D. L. Donoho, and M. Elad, "From sparse solutions of systems of equations to sparse modeling of signals and images," *SIAM Rev.*, vol. 51, no. 1, pp. 34–81, 2009.

[8] S. Ramani and J. Fessler, "Parallel MR image reconstruction using augmented Lagrangian methods," *IEEE Trans. Med. Imag.*, vol. 30, no. 3, pp. 694–706, Mar. 2011.

[9] M. Elad, M. Figueiredo, and Y. Ma, "On the role of sparse and redundant representations in image processing," *Proc. IEEE*, vol. 98, no. 6, pp. 972–982, Jun. 2010.

[10] B. Adcock and A. C. Hansen, "Generalized sampling and infinite-dimensional compressed sensing," *Found. Comput. Math.*, vol. 16, no. 5, pp. 1263–1323, 2016.

[11] B. Adcock, A. Hansen, B. Roman, and G. Teschke, "Generalized sampling: Stable reconstructions, inverse problems and compressed sensing over the continuum," in *Proc. Adv. Imag. Electron. Phys.*, 2014, pp. 187–279.

[12] A. Flinth and P. Weiss, "Exact solutions of infinite dimensional total-variation regularized problems," *Inf. Inference: A J. IMA*, vol. 8, no. 3, pp. 407–443, 2019.

[13] G. Tang, B. N. Bhaskar, P. Shah, and B. Recht, "Compressed sensing off the grid," *IEEE Trans. Inf. Theory*, vol. 59, no. 11, pp. 7465–7490, Nov. 2013.

[14] Y. C. Eldar, "Compressed sensing of analog signals in shift-invariant spaces," *IEEE Trans. Signal Process.*, vol. 57, no. 8, pp. 2986–2997, Aug. 2009.

[15] C. Poon, N. Keriven, and G. Peyré, "The geometry of off-the-grid compressed sensing," *Found. Comput. Math.*, vol. 23, no. 1, pp. 241–327, 2023.

[16] Y. Eldar and G. Kutyniok, *Compressed Sensing: Theory and Applications*. Cambridge, U.K.: Cambridge Univ. Press, 2012.

[17] B. Adcock and A. C. Hansen, *Compressive Imaging: Structure, Sampling, Learning*. Cambridge, U.K.: Cambridge Univ. Press, 2021.

[18] Y. C. Eldar, *Sampling Theory: Beyond Bandlimited Systems*. Cambridge, U.K.: Cambridge Univ. Press, 2015.

[19] H. Gupta, J. Fageot, and M. Unser, "Continuous-domain solutions of linear inverse problems with Tikhonov versus generalized TV regularization," *IEEE Trans. Signal Process.*, vol. 66, no. 17, pp. 4670–4684, Sep. 2018.

[20] M. Unser and J. Fageot, "Native banach spaces for splines and variational inverse problems," 2019, *arXiv:1904.10818*.

[21] M. Unser, J. Fageot, and J. P. Ward, "Splines are universal solutions of linear inverse problems with generalized tv regularization," *SIAM Rev.*, vol. 59, no. 4, pp. 769–793, 2017.

[22] C. Boyer, A. Chambolle, Y. D. Castro, V. Duval, F. de Gournay, and P. Weiss, "On representer theorems and convex regularization," *SIAM J. Optim.*, vol. 29, no. 2, pp. 1260–1281, 2019.

[23] R. Parhi and R. D. Nowak, "Banach space representer theorems for neural networks and ridge splines," *J. Mach. Learn. Res.*, vol. 22, no. 43, pp. 1–40, Jan. 2021.

[24] M. Unser, "A unifying representer theorem for inverse problems and machine learning," *Found. Comput. Math.*, vol. 21, no. 4, pp. 941–960, 2021.

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

[26] S. Aziznejad, J. Campos, and M. Unser, "Measuring complexity of learning schemes using hessian-schatten total variation," *SIAM J. Math. Data Sci.*, vol. 5, no. 2, pp. 422–445, 2023.

[27] A. Beck and M. Teboulle, "Fast gradient-based algorithms for constrained total variation image denoising and deblurring problems," *IEEE Trans. Image Process.*, vol. 18, no. 11, pp. 2419–2434, Nov. 2009.

[28] S. Lefkimmiatis, A. Bourquard, and M. Unser, "Hessian-based norm regularization for image restoration with biomedical applications," *IEEE Trans. Image Process.*, vol. 21, no. 3, pp. 983–995, Mar. 2012.

[29] S. Lefkimmiatis, J. P. Ward, and M. Unser, "Hessian Schatten-norm regularization for linear inverse problems," *IEEE Trans. Image Process.*, vol. 22, no. 5, pp. 1873–1888, May 2013.

[30] K. Bredies and M. Carioni, "Sparsity of solutions for variational inverse problems with finite-dimensional data," *Calculus Variations Partial Differ. Equ.*, vol. 59, no. 1, 2020, Art. no. 14.

[31] L. Ambrosio, J.-M. Morel, S. Masnou, and V. Caselles, "Connected components of sets of finite perimeter and applications to image processing," *J. Eur. Math. Soc.*, vol. 3, no. 1, pp. 39–92, 2001.

[32] L. Ambrosio, C. Brena, and S. Conti, "Functions with bounded hessian-schatten variation: Density, variational and extremality properties," 2023, *arXiv:2302.12554*.

[33] L. Ambrosio, S. Aziznejad, C. Brena, and M. Unser, "Linear inverse problems with Hessian-schatten total variation," *Calculus Variations Partial Different. Eqs.*, vol. 63, no. 1, 2024, Art. no. 9.

[34] P. Bohra and M. Unser, "Continuous-domain signal reconstruction using Lp-norm regularization," *IEEE Trans. Signal Process.*, vol. 68, pp. 4543–4554, 2020.

[35] T. Debarre, J. Fageot, H. Gupta, and M. Unser, "Solving continuous-domain problems exactly with multiresolution B-splines," in *Proc. IEEE Int. Conf. Acoust. Speech Signal Process.*, 2019, pp. 5122–5126.

[36] P. Bohra and M. Unser, "Computation of "Best" interpolants in the Lp sense," in *Proc. IEEE Int. Conf. Acoust. Speech Signal Process.*, 2020, pp. 5505–5509.

[37] T. Debarre, J. Fageot, H. Gupta, and M. Unser, "B-spline-based exact discretization of continuous-domain inverse problems with generalized tv regularization," *IEEE Trans. Inf. Theory*, vol. 65, no. 7, pp. 4457–4470, Jul. 2019.

[38] M. Unser, "Splines: A. perfect fit for signal and image processing," *IEEE Signal Process. Mag.*, vol. 16, no. 6, pp. 22–38, Nov. 1999.

[39] H. Prautzsch, W. Boehm, and M. Paluszny, "Box splines," in *Bezier B-Spline Techn.*, pp. 239–258, 2002.

[40] J. Campos, S. Aziznejad, and M. Unser, "Learning of continuous and piecewise-linear functions with essian total-variation regularization," *IEEE Open J. Signal Process.*, vol. 3, pp. 36–48, 2022.

[41] L. Condat and D. Van De Ville, "Three-directional box-splines: Characterization and efficient evaluation," *IEEE Signal Process. Lett.*, vol. 13, no. 7, pp. 417–420, Jul. 2006.

[42] A. Entezari, M. Nilchian, and M. Unser, "A box spline calculus for the discretization of computed tomography reconstruction problems," *IEEE Trans. Med. Imag.*, vol. 31, no. 8, pp. 1532–1541, Aug. 2012.

[43] A. Entezari, D. Van De Ville, and T. Moller, "Practical box splines for reconstruction on the body centered cubic lattice," *IEEE Trans. Vis. Comput. Graph.*, vol. 14, no. 2, pp. 313–328, Mar./Apr. 2008.

[44] A. Entezari and T. Moller, "Extensions of the zwart-powell box spline for volumetric data reconstruction on the cartesian lattice," *IEEE Trans. Vis. Comput. Graph.*, vol. 12, no. 5, pp. 1337–1344, Sep./Oct. 2006.

[45] M. Kim and J. Peters, "A practical box spline compendium," *Appl. Math. Comput.*, vol. 464, 2024, Art. no. 128376.

[46] A. Chambolle and T. Pock, "Chapter 6 - approximating the total variation with finite differences or finite elements," in *Geometric Partial Differential Equations - Part II* (Handbook of Numerical Analysis Series), vol. 22, A. Bonito and R. H. Nochetto, Eds. Amsterdam, The Netherlands, Elsevier, 2021, pp. 383–417.

[47] M. Herrmann, R. Herzog, S. Schmidt, J. Vidal-Núñez, and G. Wachsmuth, "Discrete total variation with finite elements and applications to imaging," *J. Math. Imag. Vis.*, vol. 61, pp. 411–431, 2019.

[48] S. Bartels, "Total variation minimization with finite elements: Convergence and iterative solution," *SIAM J. Numer. Anal.*, vol. 50, no. 3, pp. 1162–1180, 2012.

[49] G. Bellettini, V. Caselles, and M. Novaga, "The total variation flow in RN," *J. Differ. Equ.*, vol. 184, no. 2, pp. 475–525, 2002.

[50] V. Caselles, A. Chambolle, and M. Novaga, "Regularity for solutions of the total variation denoising problem," *Revista Matemática Iberoamericana*, vol. 27, no. 1, pp. 233–252, 2011.

[51] Q. Denoyelle, V. Duval, G. Peyré, and E. Soubies, "The sliding frank–wolfe algorithm and its application to super-resolution microscopy," *Inverse Problems*, vol. 36, no. 1, 2019, Art. no. 014001.

[52] Y. De Castro, V. Duval, and R. Petit, "Towards off-the-grid algorithms for total variation regularized inverse problems," *J. Math. Imag. Vis.*, vol. 65, no. 1, pp. 53–81, 2023.

[53] C. De Boor, K. Höllig, and S. Riemenschneider, *Box Splines*, vol. 98. Berlin, Germany: Springer, 2013.

[54] I. Koutromanos, *Fundamentals of Finite Element Analysis: Linear Finite Element Analysis*.Hoboken, NJ, USA: Wiley, 2018.

Authorized licensed use limited to: EPFL LAUSANNE. Downloaded on June 25,2024 at 10:28:57 UTC from IEEE Xplore. Restrictions apply.

[55] R. Arora, A. Basu, P. Mianjy, and A. Mukherjee, "Understanding deep neural networks with rectified linear units," in *Proc. Int. Conf. Learn. Representations*, Poster Presentation, 2018.
[56] M. Unser, "Approximation power of biorthogonal wavelet expansions," *IEEE Trans. Signal Process.*, vol. 44, no. 3, pp. 519–527, Mar. 1996.
[57] G. Strang and G. Fix, "A fourier analysis of the finite element variational method," *Constructive Aspects Funct. Anal.*, vol. 57, pp. 793–840, 2011.
[58] A. Goujon, J. Campos, and M. Unser, "Stable parameterization of continuous and piecewise-linear functions," *Appl. Comput. Harmon. Anal.*, vol. 67, 2023, Art. no. 101581.
[59] A. Beck and M. Teboulle, "A fast iterative shrinkage-thresholding algorithm for linear inverse problems," *SIAM Ournal Maging Ciences*, vol. 2, no. 1, pp. 183–202, 2009.
[60] M. Pourya, A. Goujon, and M. Unser, "Delaunay-triangulation-based learning with Hessian total-variation regularization," *IEEE Open J. Signal Process.*, vol. 4, pp. 167–178, 2023.
[61] S. Bonettini, S. Rebegoldi, and V. Ruggiero, "Inertial variable metric techniques for the inexact forward–backward algorithm," *SIAM J. Sci. Comput.*, vol. 40, no. 5, pp. A3180–A3210, 2018.
[62] S. Salzo et al., "Inexact and accelerated proximal point algorithms," *J. Convex Anal.*, vol. 19, no. 4, pp. 1167–1192, 2012.
[63] M. Schmidt, N. Roux, and F. Bach, "Convergence rates of inexact proximal-gradient methods for convex optimization," in *Proc. Adv. Neural Inf. Process. Syst.*, 2011.
[64] A. Chambolle and T. Pock, "A first-order primal-dual algorithm for convex problems with applications to imaging," *J. Math. Imag. Vis.*, vol. 40, pp. 120–145, 2011.
[65] L. Condat, "A primal–dual splitting method for convex optimization involving lipschitzian, proximable and linear composite terms," *J. Optim. Theory Appl.*, vol. 158, no. 2, pp. 460–479, 2013.
[66] B. C. Vũ, "A splitting algorithm for dual monotone inclusions involving cocoercive operators," *Adv. Comput. Math.*, vol. 38, no. 3, pp. 667–681, 2013.
[67] M. Yan, "A new primal–dual algorithm for minimizing the sum of three functions with a linear operator," *J. Sci. Comput.*, vol. 76, pp. 1698–1717, 2018.
[68] M.-J. Lai, B. Lucier, and J. Wang, "The convergence of a central-difference discretization of rudin-osher-fatemi model for image denoising," in *Proc. 2nd Int. Conf. Scale Space Variational Methods Comput. Vis.*, 2009, pp. 514–526.
[69] J. Wang and B. J. Lucier, "Error bounds for finite-difference methods for rudin–osher–fatemi image smoothing," *SIAM J. Numer. Anal.*, vol. 49, no. 2, pp. 845–868, 2011.
[70] H. Alvis-Miranda, S. Castellar Leones, G. Alcalá-Cerra, and L. Moscote-Salazar, "Cerebral sinus venous thrombosis," *J. Rural Pract.*, vol. 4, pp. 427–438, 2013.
[71] K. Egiazarian, A. Foi, and V. Katkovnik, "Compressed sensing image reconstruction via recursive spatially adaptive filtering," in *Proc. IEEE Int. Conf. Image Process.*, 2007, pp. I-549–I-552.
[72] A. Chambolle, S. E. Levine, and B. J. Lucier, "An upwind finite-difference method for total variation–based image smoothing," *SIAM J. Imag. Sci.*, vol. 4, no. 1, pp. 277–299, 2011.
[73] L. Condat, "Discrete total variation: New definition and minimization," *SIAM J. Imag. Sci.*, vol. 10, no. 3, pp. 1258–1290, 2017.

Mehrsa Pourya (Graduate Student Member, IEEE) received the B.Sc. degree in electrical engineering from the Sharif University of Technology, Tehran, Iran, in 2020. She is currently working toward the Ph.D. degree with the Biomedical Imaging Group, École Polytechnique Fédérale de Lausanne, Lausanne, Switzerland, under the direction of Michael Unser. Her research interests include signal processing and machine learning.

Aleix Boquet-Pujadas received the bachelor's degree in mathematics and the bachelor's degree in physics from Universitat Autònoma de Barcelona (first-in-class), Bellaterra, Spain, and the Ph.D. degree from Sorbonne Université (or UPMC, or Paris VI), Paris, France. His doctoral thesis was supported by a Marie Skłodowska-Curie fellowship at Institut Pasteur and distinguished by the French Society of Biomedical Engineering (SFGBM) for its innovation. He is currently a Postdoctoral Researcher with the Biomedical Imaging Group, EPFL.

Michael Unser (Fellow, IEEE) received the M.S. (summa cum laude) and Ph.D. degrees in electrical engineering from Ecole Polytechnique Fédérale de Lausanne (EPFL), Lausanne, Switzerland, in 1981 and 1984, respectively. He is currently a Professor and the Director of EPFL's Biomedical Imaging Group, Lausanne. He is the author with P. Tafti of the book An introduction to sparse stochastic processes Cambridge University Press 2014. He has authored or coauthored more than 350 journal papers in his research areas which include sampling theory, wavelets, the use of splines for image processing, stochastic processes, and computational bioimaging. His primary area of investigation is biomedical image processing.

From 1985 to 1997, he was with the Biomedical Engineering and Instrumentation Program, National Institutes of Health, Bethesda USA, conducting research on bioimaging.

He was an Associate Editor-in-Chief for most of the primary journals in his field including IEEE TRANSACTIONS ON MEDICAL IMAGINGduring 2003–2005, IEEE TRANSACTIONS ON IMAGE PROCESSING, PROCEEDINGS OF IEEE, and *SIAM Journal of Imaging Sciences*. He is the founding Chair of the technical committee on Bio Imaging and Signal Processing of the IEEE Signal Processing Society.

He is an EURASIP Fellow (2009) and a Member of the Swiss Academy of Engineering Sciences. He was the recipient of several international prizes including five IEEE-SPS Best Paper Awards, two Technical Achievement Awards from the IEEE (2008 SPS and EMBS 2010), Technical Achievement Award from EURASIP (2018), and recently Career Achievement Award (IEEE EMBS 2020).

Authorized licensed use limited to: EPFL LAUSANNE. Downloaded on June 25,2024 at 10:28:57 UTC from IEEE Xplore. Restrictions apply.