Discrete denoising of heterogenous two-dimensional data

Taesup Moon[1], Tsachy Weissman[2], and Jae-Young Kim[2]

[1]Yahoo! Labs, 701 First Ave, Sunnyvale, CA 94089
[2]Stanford University, 350 Serra Mall, Stanford, CA 94305

July 13, 2010

Abstract

We consider discrete denoising of two-dimensional data with characteristics that may be varying abruptly between regions. Using a quadtree decomposition technique and space-filling curves, we extend the recently developed S-DUDE (Shifting Discrete Universal DEnoiser), which was tailored to one-dimensional data, to the two-dimensional case. Our scheme competes with a genie that has access, in addition to the noisy data, also to the underlying noiseless data, and can employ m different two-dimensional sliding window denoisers along m distinct regions obtained by a quadtree decomposition with m leaves, in a way that minimizes the overall loss. We show that, regardless of what the underlying noiseless data may be, the two-dimensional S-DUDE performs essentially as well as this genie, provided that the number of distinct regions satisfies $m = o(n)$, where n is the total size of the data. The resulting algorithm complexity is still linear in both n and m, as in the one-dimensional case. Our experimental results show that the two-dimensional S-DUDE can be effective when the characteristics of the underlying clean image vary across different regions in the data.

Index Terms- discrete denoising, two-dimensional data, quadtree decomposition, space-filling curves, Peano-Hilbert scan

1 Introduction

Discrete denoising is the problem of reconstructing the components of a finite-alphabet sequence based on the observation of its Discrete Memoryless Channel (DMC)[1]-corrupted version. Universal discrete denoising, in which no statistical or other properties are known a priori about the underlying clean data and the goal is to attain optimum performance, was considered and solved in [1]. The main result in [1] is the *semi-stochastic* setting one, which asserts that, regardless of what the underlying individual sequence may be, the Discrete Universal Denoiser (DUDE) attains the performance of the *best* sliding window denoiser that would be chosen by a genie who accesses, in addition to the noisy sequence, the underlying clean data. Recently, in [2], a generalization has been carried out for the case in which the characteristics of the underlying sequence change over time. The new scheme, called Shifting Discrete Universal Denoiser (S-DUDE), was shown to achieve the performance of the *best combination* of sliding window denoisers, allowing at most m shifts (i.e., switches from one sliding window denoiser to another) along the sequence, provided that m grows sub-linearly in the data size n, regardless of what the underlying noiseless sequence may be. It was also shown in [2] that the scheme can be implemented efficiently via dynamic programming, with linear complexity both in n and m.

One of the domains in which DUDE found its application is image denoising. It was shown in the experimental results of [1] and [3] that DUDE achieves or often outperforms the best of several of the state-of-the-art image denoisers for small-alphabet images, many of which are sliding window schemes. It is natural then to attempt to extend S-DUDE for images, namely, two-dimensional data, as well. The motivation is clear; images tend to have locally distinct characteristics, and allowing the sliding window denoisers to shift from one region to another may significantly improve the denoising performance compared to applying one fixed sliding window denoiser throughout

[1]The DMC is assumed known throughout this paper. This assumption is benign in applications where the DMC is easily learnable from the data.

all the data. However, whereas the extension of the DUDE to two-dimensional data was straightforward (cf. [1, Section VIII-C] and [3]), that of the S-DUDE is highly non-trivial, since it requires segmentation of the data, based on its noisy observation, into homogeneous regions in a way that minimizes the overall loss. Such segmentation is significantly more involved and often intractable, in contrast to the one-dimensional case of the S-DUDE, which only required to divide the data into distinct intervals with associated denoisers.

Due to this difficulty of general segmentation of data, we instead adopt a restricted, yet rich enough, segmentation scheme - quadtree decomposition - to build a reference class of shifting two-dimensional sliding window denoisers. Then, we employ the space-filling Peano-Hilbert curve [4, 5] to scan the data so that applying the original one-dimensional S-DUDE on the scanned data can achieve the best performance among the schemes in the reference class, regardless of the underlying clean data. The quadtree decomposition has been popular in image compression [6][7] and pattern recognition [8], and recently in [9], it has also been applied to denoising continuous-valued signals by viewing denoising as a low-rate lossy compression problem. The Peano-Hilbert curves have been used, among other applications, in universal compression of two-dimensional data in both the individual sequence setting [4] and the probabilistic setting [10]. A more general problem of scanning and predicting multi-dimensional data was considered in [11, 12]. The combination of the quadtree decomposition and the Peano-Hilbert scanning for discrete denoising problems is the main contribution of this paper.

Our resulting denoising scheme, 2-D S-DUDE, still enjoys the performance guarantees that parallel those of [2] for two-dimensional data. That is, regardless of what the underlying clean data might be, 2-D S-DUDE performs asymptotically as well as the best combination of the two-dimensional sliding window denoisers that can shift across at most m distinct regions, as can be segmented by the best quadtree decomposition. Our use of the Peano-Hilbert scan is essential to obtain a scheme of which complexity remains linear in both the data size and the number of distinct segments m, whereas an effort of directly finding the best quadtree decomposition may have resulted in a scheme with complexity exponential in m. We show the effectiveness of our scheme by experimental results that demonstrate 2-D S-DUDE outperforming 2-D DUDE, particularly for images of space-varying characteristics, such as scanned magazines, etc.

The rest of the paper is organized as follows: Section 2 collects necessary notation, preliminary results and detailed motivation for this work. Our main theoretical results and algorithm are given in Section 3, and the experimental results are presented in Section 4. Concluding remarks with a discussion of future work are given in Section 5.

2 Notation, Preliminaries, and Motivation

2.1 Notation

We follow the notation of [2]. Let $\mathcal{X}, \mathcal{Z}, \hat{\mathcal{X}}$ denote, respectively, the alphabet of the clean, noisy, and reconstructed sources, which are assumed to be finite. As in [1, 2, 13], the noisy sequence is a DMC-corrupted version of the clean one, where the channel matrix is denoted by $\mathbf{\Pi} = \{\Pi(x,z)\}_{x\in\mathcal{X},z\in\mathcal{Z}}$, and $\Pi(x,z)$ stands for the probability of a noisy symbol z when the underlying clean symbol is x. Throughout the paper, $\mathbf{\Pi}$ is assumed to be known and fixed, and of full row rank. When a reconstruction $\hat{x}$ is made for a clean symbol x, the goodness of the reconstruction is measured by a loss function $\Lambda : \mathcal{X} \times \hat{\mathcal{X}} \to [0,\infty)$. Upper case letters denote random variables; lower case letters denote either individual deterministic quantities or specific realizations of random variables. Without loss of generality, the elements of any finite alphabet $\mathcal{V}$ will be identified with $\{0, 1, \cdots, |\mathcal{V}|-1\}$. For $\mathcal{V}$-valued sequence, we let $v^n = (v_1, \cdots, v_n)$, $v_m^n = (v_m, \cdots, v_n)$, and $v^{n\backslash t} = v^{t-1}v_{t+1}^n$. $\mathbb{R}^{\mathcal{V}}$ is a space of $|\mathcal{V}|$-dimensional column vectors with real-valued components indexed by the elements of $\mathcal{V}$.

Now, consider the set $\mathcal{S} = \{s : \mathcal{Z} \to \hat{\mathcal{X}}\}$, which is the (finite) set of mappings that take $\mathcal{Z}$ into $\hat{\mathcal{X}}$. We refer to elements of $\mathcal{S}$ as "single-symbol denoisers", since each $s \in \mathcal{S}$ can be thought of as a rule for estimating $x \in \mathcal{X}$ on the basis of $z \in \mathcal{Z}$. Then, for any $s \in \mathcal{S}$, we can always devise an estimated loss $\ell(Z, s)$ with the knowledge of $\mathbf{\Pi}$, which is an unbiased estimate of the true expected loss $E_x\Lambda(x, s(Z))$, i.e. satisfying

$$E_x\ell(Z,s) = E_x\Lambda(x,s(Z)) \quad \forall x \in \mathcal{X}. \tag{1}$$

The expectation in (1) is with respect to the conditional distribution on Z given x, $\Pi(x,\cdot)$. For more details on the motivation for using a loss function ℓ satisfying (1), and on its explicit form, readers may refer to [2, Section II-A].

2.2 DUDE and S-DUDE for 1-D data

Here, we review and summarize the results from [1] and [2], and collect the ideas that will be needed for this paper. For one dimensional data, an *n-block denoiser* is a collection of n mappings $\hat{\mathbf{X}}^n = \{\hat{X}_t\}_{1\le t\le n}$, where $\hat{X}_t : \mathcal{Z}^n \to \hat{\mathcal{X}}$. The performance of the denoiser $\hat{\mathbf{X}}^n$ on the individual sequence pair (x^n, z^n) is measured by its normalized cumulative loss

$$L_{\hat{\mathbf{X}}^n}(x^n, z^n) = \frac{1}{n}\sum_{t=1}^{n} \Lambda(x_t, \hat{X}_t(z^n)).$$

As argued in [2, Section II-B], the n-block denoiser $\hat{\mathbf{X}}^n = \{\hat{X}_t\}_{1\le t\le n}$ can be identified with $\mathbf{F}^n = \{F_t\}_{1\le t\le n}$, where $F_t : \mathcal{Z}^{n\setminus t} \to \mathcal{S}$ is defined as follows: $F_t(z^{n\setminus t}, \cdot)$ is the single-symbol denoiser in $\mathcal{S}$ satisfying

$$\hat{X}_t(z^n) = F_t(z^{n\setminus t}, z_t), \quad \forall z_t. \tag{2}$$

One special class of widely used n-block denoisers is that of the k-th order "sliding window" denoisers, which we denote by $\hat{\mathbf{X}}^{n,\mathcal{S}_k}$. Such denoisers are of the form

$$\hat{X}_t^{s_k}(z^n) = s_k(z_{t-k}^{t+k}) = s_k(\mathbf{c}_t, z_t) \tag{3}$$

for $t = k+1, \cdots, n-k$, where s_k is an element of $\mathcal{S}_k = \{s_k : \mathcal{Z}^{2k+1} \to \hat{\mathcal{X}}\}$, the (finite) set of mappings from $\mathcal{Z}^{2k+1}$ into $\hat{\mathcal{X}}$, and $\mathbf{c}_t \triangleq (z_{t-k}^{t-1}, z_{t+1}^{t+k})$ is the (two-sided) k-th order context for z_t. We refer to $s_k \in \mathcal{S}_k$ as a "k-th order denoiser". Note that $\mathcal{S}_0$ equals to $\mathcal{S}$ in the previous section. $\mathbf{C}_k \triangleq \{(u_{-k}^{-1}, u_1^k) : (u_{-k}^{-1}, u_1^k) \in \mathcal{Z}^{2k}\}$ is the set of all possible k-th order contexts, and for given z^n and each $\mathbf{c} \in \mathbf{C}_k$,

$$\mathcal{T}(\mathbf{c}) \triangleq \left\{\tau : \mathbf{c}_\tau = \mathbf{c}, \quad k+1 \le \tau \le n-k\right\}$$

is further defined to be the set of indices where the context of z_i equals $\mathbf{c}$. From the association in (2) and the definition (3), we can deduce that for each $\mathbf{c} \in \mathbf{C}_k$, the k-th order sliding window denoiser employs a time-invariant single-symbol denoiser, $s_k(\mathbf{c}, \cdot)$, at all points $t \in \mathcal{T}(\mathbf{c})$.

In [1], the performance target of the denoising is

$$D_k(x^n, z^n) \triangleq \min_{s_k \in \mathcal{S}_k} \frac{1}{n-2k} \sum_{t=k+1}^{n-k} \Lambda(x_t, s_k(\mathbf{c}_t, z_t)),$$

the minimum normalized loss on (x^n, z^n) that can be attained by a k-th order sliding window denoiser. We can easily verify that for each $\mathbf{c} \in \mathbf{C}_k$, the best k-th order sliding window denoiser that achieves $D_k(x^n, z^n)$ will employ the single-symbol denoiser

$$\arg\min_{s\in\mathcal{S}} \sum_{\tau\in\mathcal{T}(\mathbf{c})} \Lambda(x_\tau, s(z_\tau)), \tag{4}$$

at all points $t \in \mathcal{T}(\mathbf{c})$, which is determined from the joint empirical distribution of pairs $\{(x_\tau, z_\tau) : \tau \in \mathcal{T}(\mathbf{c})\}$. It is shown in [1, Theorem 1] that, despite the lack of knowledge of x^n, $D_k(x^n, Z^n)$ is essentially achievable by the Discrete Universal DEnoiser (DUDE), which accesses only Z^n and is implementable with linear complexity in n. For each $\mathbf{c} \in \mathbf{C}_k$, the DUDE can be shown to employ

$$\arg\min_{s\in\mathcal{S}} \sum_{\tau\in\mathcal{T}(\mathbf{c})} \ell(z_\tau, s) \tag{5}$$

at all points $t \in \mathcal{T}(\mathbf{c})$, where $\ell(z, s)$ is the loss function chosen to satisfy (1). By comparing (4) with (5), we observe that, for each context, the DUDE merely works with the estimated loss $\ell(z_\tau, s)$ in lieu of the true loss $\Lambda(x_\tau, s(z_\tau))$.

The idea of working with estimated loss to achieve the genie-aided performance has been adopted again in [2] to refine the result of [1]. The main motivation of [2] was the observation that when the characteristics of the underlying sequence x^n change with time, allowing the k-th order denoiser to change from one interval of the data to another can further reduce the overall loss significantly. Therefore, whereas the DUDE competed with the best *fixed* k-th

order denoiser, [2] competes with the best among $\mathcal{S}^n_{k,m}$, a set of "combinations" of k-th order denoisers $\{s_{k,t}\}_{t=k+1}^{n-k}$ that allow at most m shifts within $t \in \mathcal{T}(\mathbf{c})$ for each $\mathbf{c} \in \mathbf{C}_k$. Thus, [2] sets a more ambitious performance target

$$D_{k,m}(x^n, z^n) \triangleq \min_{\mathbf{s} \in \mathcal{S}^n_{k,m}} \frac{1}{n-2k} \sum_{t=k+1}^{n-k} \Lambda(x_t, s_{k,t}(\mathbf{c}_t, z_t)), \tag{6}$$

the minimum normalized loss on (x^n, z^n) that can be achieved by the sequence of k-th order denoisers that allow at most m shifts (changes) within each context. It is clear that $D_{k,m}(x^n, z^n) \leq D_k(x^n, z^n)$ for all (x^n, z^n). The new algorithm devised in [2] was called the (k, m)-Shifting Discrete Universal DEnoiser (S-DUDE), $\hat{\mathbf{X}}^{n,k,m}_{\text{univ}}$, and was able to asymptotically achieve $D_{k,m}(x^n, Z^n)$ on the basis of Z^n only, provided that m grows sub-linearly in n. The key trick was again to work with the estimated loss $\ell(z, s)$ to obtain

$$\hat{\mathbf{S}}_{k,m} = \arg \min_{\mathbf{s} \in \mathcal{S}^n_{k,m}} \frac{1}{n-2k} \sum_{t=k+1}^{n-k} \ell(z_t, s_{k,t}(\mathbf{c}_t, \cdot)), \tag{7}$$

and employ it throughout the sequence. The following theorem shows that by utilizing the estimated loss, we can successfully learn the best (at most m) shifts of k-th order denoisers throughout z^n to minimize the overall normalized loss.

Theorem 1 *([2, Theorem 4]) Suppose $k = k_n$ and $m = m_n$ grow with n sufficiently slowly to satisfy conditions detailed in [2, Claim 1]. Then, for all $\mathbf{x} \in \mathcal{X}^\infty$, the sequence of denoisers $\{\hat{\mathbf{X}}^{n,k,m}_{univ}\}$ satisfies*

a) $\lim_{n\to\infty} \left[L_{\hat{\mathbf{X}}^{n,k,m}_{\text{univ}}}(x^n, Z^n) - D_{k,m}(x^n, Z^n) \right] = 0 \quad a.s.$

b) For any $\delta > 0$, $E[L_{\hat{\mathbf{X}}^{n,k,m}_{\text{univ}}}(x^n, Z^n) - D_{k,m}(x^n, Z^n)] = O\big(\sqrt{k_n |\mathcal{Z}|^{2k_n} (\frac{m_n}{n})^{1-\delta}}\big)$.

Besides the performance guarantees, another key component of [2] was developing an algorithm that can implement (7) efficiently, i.e., with complexity linear in both n and m. The details can be found in [2, Section V-A].

2.3 Motivation

As pointed out in [2, Section V-B], both DUDE and S-DUDE run the same algorithm in parallel along each subsequence associated with each context, and this characteristic enables us to extend both schemes to the two-dimensional (2-D) data case: use 2-D contexts and again run the algorithms on each subsequence. However, as noted in the Introduction, the extension to 2-D data of the S-DUDE is not as straightforward as that of the DUDE. The main reason is that, whereas the output of DUDE is independent of the ordering of data within each context and only requires the empirical distribution of the data, the ordering of said data is very consequential for S-DUDE's output and its performance.

The ordering of data is naturally given and fixed in one-dimensional (1-D) data. Therefore, S-DUDE only had to find shifting points based on noisy data so that applying different sliding window denoisers in different intervals will minimize the overall loss. Figure 1 shows one such segmentation in which different colors represent intervals where different sliding window denoisers are applied. In contrast, in the 2-D case, it is not clear how the 2-D version

Figure 1: Segmentation of 1-D data

of S-DUDE should segment into *homogeneous* 2-D regions, instead of intervals, in order to allow shifting of sliding window denoisers across the data. As depcited in Figure 2, the optimal segmentation that leads to the minimum loss can be arbitrary, and hence, trying to learn the best segmentation solely based on *noisy* data would be overly ambitious. One naive approach to avoid such a 2-D segmentation issue would be to first raster scan the image, then

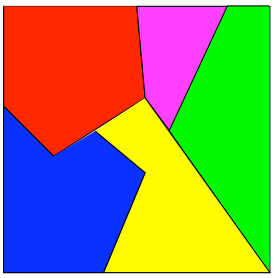


Figure 2: Segmentation of 2-D data

apply the ordinary S-DUDE on the resulting 1-D data, which was the method used in [2, Section VI-A]. However, this could often result in poor performance of the scheme since it may require the S-DUDE to shift too frequently, i.e., m to become linear in the data size n, which violates the necessary condition specified in [2, Theorem 5] for the scheme to work. This point can be seen by imagining the situation of running the raster scan vertically on the image in [2, Figure 2], where even though the image consists of a small number of two-dimensional regions, when raster scanned into one-dimensional data the number of changes of the data characteristics grows significantly.

To address this issue of segmenting and scanning of the 2-D data in general, in this paper we focus on a more regularized class of segmentation schemes, the quadtree decomposition, to build the reference class of shifting sliding window denoisers rich enough for the denoising task at hand. Then, in order to compete with the reference class, we utilize the space-filling Peano-Hilbert (PH) curve to scan subsequence points for each 2-D context and run the ordinary S-DUDE on the P-H scanned, 1-D data. In the next section, we describe our scheme formally, and prove a performance guarantee for it, which parallels that of the scheme for 1D-data in [2].

3 S-DUDE for 2-D data

Before presenting the 2-D extension of S-DUDE, we introduce additional notation in Section 3.1 through Section 3.3. Then, in Section 3.4, we derive our scheme and present theoretical guarantees of its performance. In Section 3.5, we succinctly describe the algorithm and its complexity.

3.1 2-D data and contexts

We represent the 2-D data with the coordinate of each data point. For simplicity, we assume the 2-D data is always in the square form[2]. Then, denote $\mathcal{T}_N \triangleq \{t \in \mathbb{Z}^2 : t = (t_1, t_2), 1 \le t_1 \le N, 1 \le t_2 \le N\}$ as the set of coordinates of the given 2-D data. Also, let $n = |\mathcal{T}_N| = N \times N$ be the total size of the data. For $t \in \mathcal{T}_N$, z_t will denote the noisy symbol at location $t = (t_1, t_2)$, and $x^{N\times N}$ and $z^{N\times N}$ will denote the total clean and noisy 2-D data, respectively. In addition, for $t \in \mathcal{T}_N$, $z^{N\times N\setminus t}$ will denote $\{z_i : i \in \mathcal{T}_N, i \neq t\}$. With this notation, notions of the 2-D n-block denoiser $\hat{\mathbf{X}}^n_{2D}$, the normalized cumulative loss

$$L_{\hat{\mathbf{X}}^n_{2D}}(x^{N\times N}, z^{N\times N}) = \frac{1}{n}\sum_{t\in\mathcal{T}_N} \Lambda(x_t, \hat{X}_{t,2\mathrm{D}}(z^{N\times N})),$$

and the association in (2) follow naturally in parallel to the 1-D data case.

The 2-D k-th order sliding window denoisers can be understood similarly. First, consider the sequence of coordinates, $\mathcal{I} = (i_1, i_2, i_3, \cdots)$, in the 2-D lattice of integers, in which the coordinates are enumerated in the order of increasing distance to the origin as in Figure 3. Then, 2-D k-th order context for z_t is defined to be $\mathbf{c}_{t,2D} = (z_{t+i_1}, \cdots, z_{t+i_{2k}})$, where $(i_1, \cdots, i_{2k})$ are the first $2k$ coordinates of $\mathcal{I}$, and the additions of coordinates simply boil down to the translation of the coordinates. We also denote $\mathbf{C}_{k,2\mathrm{D}}$ as a set of all possible 2-D k-th order contexts. Then, 2-D k-th order sliding window denoiser at location t is again of the form

$$\hat{X}^{s_k}_{t,2\mathrm{D}}(z^n) = s_k(\mathbf{c}_{t,2\mathrm{D}}, z_t),$$

with $\mathbf{c}_{t,2\mathrm{D}} \in \mathbf{C}_{k,2\mathrm{D}}$, parallelling (3).

[2]For other cases, we can simply fill in remaining regions with dummy symbols.

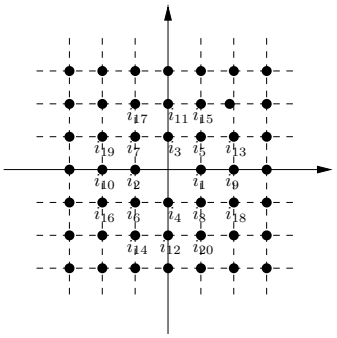
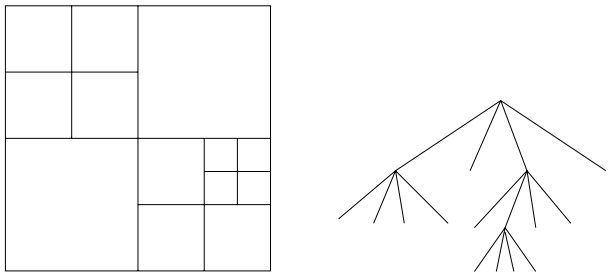


Figure 3: The ordering coordinates of the 2-D integer lattice plane

3.2 The quadtree (QT) decomposition

A quadtree (QT) is a tree of which every node is either a leaf or a parent node with four children. This QT structure can be used to segment the 2-D data as follows. Each node of a QT at depth d represents a quadrant of size $\frac{N}{2^d} \times \frac{N}{2^d}$ (assuming $N = 2^r$ and $d \leq r$), and a child node represents one of the four quadrants inside the parent node's quadrant. The four children of a parent node associate with the four quadrants of the parent node's quadrant in the order of (upper-left quadrant)→(upper-right quadrant)→(lower-right quadrant)→(lower-left quadrant). Obviously, the root node is associated with the whole 2-D data. The leaves of a QT represent the final segmentation of the 2-D data for given QT. Thus, if a QT has m leaves, the resulting segmentation has m distinct regions. For example, in Figure 4(a), the two dimensional data is segmented into 13 different regions, and the corresponding QT in Figure 4(b) has $m = 13$ leaves. We also denote Q_m as a set of all QTs that have m leaves, and for given $q \in Q_m$,



(a) 2-D data decomposition (b) QT decomposition

Figure 4: The decomposition of a 2-D data and its corresponding QT ($m = 13$)

define $R_q : \mathcal{T}_N \to \{1, \cdots, m\}$ as a mapping that maps each coordinate of the 2-D data to the one of m leaves in q corresponding to the region containing the coordinate. Although the QT decomposition is limited in the sense that it only decomposes the data into quadrants, it still is shown to be effective in many applications since it gives a compact representation of segmentation and is rich enough for capturing local similarity of data.

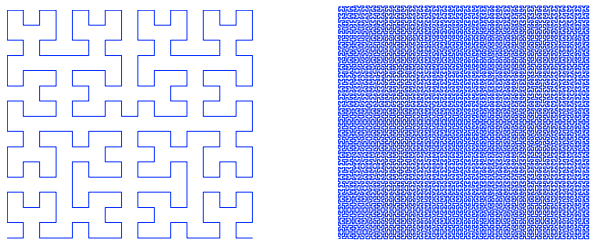
3.3 Peano-Hilbert (PH) Curve

The Peano-Hilbert (PH) curves are well known as space-filling curves. They possess the property that, for any level of quadrants, the curves never leave a quadrant before visiting all the sites within the quadrant. The details of PH curves and scanning orders can be found in [4, Section II]. Examples of PH curves for 2-D data with $N = 2^4$ and $N = 2^8$ are shown in Figure 5.

The PH curve naturally defines an ordering of 2-D according to the order in which the PH curve fills the plane. Then, for noisy 2-D data $z^{N \times N}$, we denote z^n_{PH} as its PH scanned noisy 1-D sequence and x^n_{PH} as the corresponding clean 1-D sequence. In addition, we denote the ith index according to the PH scan by PH_i. Note that

$$\mathrm{PH}_i \in \mathcal{T}_N, \quad i = 1 \ldots, n.$$

Thus, for example, z_{PH_i} stands for the ith component of the n-tuple z^n_{PH}, and $\mathbf{c}_{\mathrm{PH}_i,2\mathrm{D}} \in \mathbf{C}_{k,2\mathrm{D}}$ is the 2-D k-th order context at that location.



(a) PH(4) curve (16×16) (b) PH(8) curve (256×256)

Figure 5: Peano-Hilbert curves

3.4 Derivation of the scheme and performance guarantees

Equipped with the definitions and notation in the previous subsections, we now follow the line of argument paralleling that in [2]: begin with the case of symbol-by-symbol denoisers ($k = 0$) to build the main idea, then move on to the general k-th order case. For simplicity, we assume $N = 2^r$, and $n = N \times N = 2^{2r}$.

3.4.1 Competing with Combinations of Single-Symbol Denoisers ($k = 0$)

Consider a 2-D n tuple of single-symbol denoisers $\mathbf{S} = \{s_t : t \in \mathcal{T}_N\} \in \mathcal{S}_0^n$. For such $\mathbf{S}$, we can associate the 2-D n-block denoiser $\hat{\mathbf{X}}_{2\mathrm{D}}^{n,\mathbf{S}}$ as

$$\hat{X}^{\mathbf{S}}_{t,2\mathrm{D}}(z^{N\times N}) = s_t(z_t) \tag{8}$$

for all $t \in \mathcal{T}_N$. In order to construct the reference class based on QT decomposition described in Section 2.3, we define a subset $\mathbf{S}_m(q) \subset \mathcal{S}_0^n$ associated with a given QT $q \in Q_m$ as

$$\mathbf{S}_m(q) \triangleq \{\mathbf{S} \in \mathcal{S}_0^n : s_i = s_j \text{ if } R_q(i) = R_q(j), \quad \text{for all} \quad i, j \in \mathcal{T}_N\}.$$

In words, $\mathbf{S}_m(q)$ is a set of 2-D n-tuples of single-symbol denoisers, of which denoising rules are constant within each of the m distinct regions defined by a QT $q \in Q_m$. Now, for a fixed n and m, define a set $\mathcal{S}^n_{0,Q_m} \subset \mathcal{S}_0^n$ as

$$\mathcal{S}^n_{0,Q_m} \triangleq \bigcup_{q\in Q_m} \mathbf{S}_m(q), \tag{9}$$

which is a set of all possible configurations of single-symbol denoisers confined to be constant in regions defined by QTs with m leaves. Following is a simple lemma presenting a lower bound on the size of the set $\mathcal{S}^n_{0,Q_m}$ in terms of the number of segmented regions m.

Lemma 1 *The set $\mathcal{S}^n_{0,Q_m}$ defined in (9) satisfies*

$$|\mathcal{S}^n_{0,Q_m}| = \Omega\big(3^{\frac{m}{3}}\big).$$

Proof: By the definition of the QT, we observe that the number of leaves always has the form of $m = 3j+1$, where $j \in \mathbb{N} \cup \{0\}$. Then, for given m, we can see that

$$|\mathcal{S}^n_{0,Q_m}| \quad \geq \quad \prod_{i=0}^{j-1}(3i+1) \geq 3^{\frac{m-4}{3}},$$

which is from the fact that a new segmentation of a quadrant can happen in any of the leaves of the original QT. Therefore,

$$\lim_{m\to\infty} \frac{|\mathcal{S}^n_{0,Q_m}|}{3^{m/3}} > 0,$$

and we have the lemma. ■

Given the reference class of switching single-symbol denoisers (9), we define the performance target for given 2-D data $(x^{N\times N}, z^{N\times N})$ as

$$D_{0,m}^{2D}(x^{N\times N}, z^{N\times N}) = \min_{\mathbf{S}\in\mathcal{S}_{0,Q_m}^n} \frac{1}{n}\sum_{t\in\mathcal{T}_N} \Lambda(x_t, s_t(z_t)), \tag{10}$$

i.e., the best denoising performance attainable among all combinations of single symbol denoisers in $\mathcal{S}_{0,Q_m}^n$. In order to find the combination of single-symbol denoisers that asymptotically achieve (10) based only on $Z^{N\times N}$, we may again use the idea of utilizing the estimated loss (1) in place of the true loss as in [2] to find

$$\arg\min_{\mathbf{S}\in\mathcal{S}_{0,Q_m}^n} \frac{1}{n}\sum_{t\in\mathcal{T}_N} \ell(Z_t, s_t). \tag{11}$$

However, the naive brute-force algorithm to find the achiever of (11) requires the exhaustive search over the set $\mathcal{S}_{0,Q_m}^n$, which results in *exponential* complexity in m as specified in Lemma 1. In contrast to the 1-D case, an efficient algorithm that directly finds the best combination of single-symbol denoisers in $\mathcal{S}_{0,Q_m}^n$ does not appear to exist. To circumvent this issue, the PH scanning comes into play and serves as a key component for devising an efficient algorithm to attain performance essentially at least as good as (10). To this end, we define another set of combinations of single-symbol desnoisers

$$\mathcal{S}_{0,m}^{\mathrm{PH}(n)} \triangleq \big\{\mathbf{S}\in\mathcal{S}_0^n : \sum_{i=1}^n \mathbf{1}_{\{s_{\mathrm{PH}_{i-1}}\neq s_{\mathrm{PH}_i}\}} \le m\big\}. \tag{12}$$

In words, $\mathcal{S}_{0,m}^{\mathrm{PH}(n)}$ is a set of combinations of single-symbole denoisers that have at most m switches when the denoisers are ordered according to the PH scanning order. Equation (12) is identical to [2, eq. (20)] except that it is for the PH scanned 1-D data of the original 2-D data. We can now define

$$\hat{\mathbf{S}} = \arg\min_{\mathbf{S}\in\mathcal{S}_{0,m}^{PH(n)}} \frac{1}{n}\sum_{i=1}^n \ell(Z_{\mathrm{PH}_i}, s_{\mathrm{PH}_i}), \tag{13}$$

which can be found with *linear* complexity both in n and m by the two-pass dynamic programming algorithm established in [2, Section V]. We denote our 2-D $(0,m)$-S-DUDE as $\hat{\mathbf{X}}_{\mathrm{2D\ univ}}^{n,0,m}$, and define it to be $\hat{\mathbf{X}}_{\mathrm{2D}}^{n,\hat{\mathbf{S}}}$ (recalling the notation $\hat{\mathbf{X}}_{\mathrm{2D}}^{n,\mathbf{S}}$ from (8)). Before stating the performance guarantee of our scheme, we have following lemma.

Lemma 2 *Define a quantity*

$$D_{0,m}^{PH}(x^{N\times N}, z^{N\times N}) \triangleq \min_{\mathbf{S}\in\mathcal{S}_{0,m}^{PH(n)}} \frac{1}{n}\sum_{t\in\mathcal{T}_N} \Lambda(x_t, s_t(Z_t)).$$

Then, we have

$$D_{0,m}^{PH}(x^{N\times N}, Z^{N\times N}) \le D_{0,m}^{2D}(x^{N\times N}, z^{N\times N}).$$

for every $(x^{N\times N}, z^{N\times N})$.

Proof: From the definitions (9) and (12), we can see that

$$\mathcal{S}_{0,Q_m}^n \subseteq \mathcal{S}_{0,m}^{PH(n)},$$

since any 2-D n-tuple single-symbol denoisers in $\mathcal{S}_{0,Q_m}^n$ would be also in $\mathcal{S}_{0,m}^{PH(n)}$ after reordering them in the PH scanning order. This is because PH scan never leaves a quadrant before visiting all the data points in a quadrant, and the shifting positions for sequences in $\mathcal{S}_{0,m}^{PH(n)}$ can appear anywhere in the data, whereas the shifting positions in $\mathcal{S}_{0,Q_m}^n$ always appear on the boundary of the quadtree decomposed quadrants. Therefore, as the objective functions are identical for $D_{0,m}^{\mathrm{PH}}(x_{\mathrm{PH}}^n, z_{\mathrm{PH}}^n)$ and $D_{0,m}^{\mathrm{2D}}(x^{N\times N}, z^{N\times N})$, we get the lemma. ■

The following theorem gives the concentration result on the performance of $\hat{\mathbf{X}}_{\mathrm{2D\ univ}}^{n,0,m}$.

Theorem 2 *For $\hat{\mathbf{X}}^{n,0,m}_{2D\ univ}$ defined in (13), and for all $\epsilon > 0$ and $x^{N\times N} \in \mathcal{X}^{N\times N}$, we have*

$$\begin{aligned}&\Pr\Big(L_{\hat{\mathbf{X}}^{n,0,m}_{2D\ univ}}(x^{N\times N}, Z^{N\times N}) - D^{2D}_{0,m}(x^{N\times N}, Z^{N\times N}) > \epsilon\Big)\\ &\le 2\exp\Big(-n\Big[\frac{\epsilon^2}{2L^2_{\max}} - 2\Big\{h\Big(\frac{m}{n}\Big) + \frac{(m+1)\ln|\mathcal{S}|}{n}\Big\}\Big]\Big)\end{aligned} \tag{14}$$

where $h(x) = -x\ln x - (1-x)\ln(1-x)$ for $0 \le x \le 1$, $L_{\max} = \max_{x\in\mathcal{X},z\in\mathcal{Z}}\Lambda(x,z) + \max_{z\in\mathcal{Z},s\in\mathcal{S}}\ell(z,s)$, and $|\mathcal{S}| = |\hat{\mathcal{X}}|^{|\mathcal{Z}|}$, the size of the set of single-symbol denoisers.

Proof: From the union bound, we have

$$\begin{aligned}&\Pr\Big(L_{\hat{\mathbf{X}}^{n,0,m}_{2\mathrm{D\ univ}}}(x^{N\times N}, Z^{N\times N}) - D^{2\mathrm{D}}_{0,m}(x^{N\times N}, Z^{N\times N}) > \epsilon\Big)\\ &= \Pr\Big(L_{\hat{\mathbf{X}}^{n,0,m}_{2\mathrm{D\ univ}}}(x^{N\times N}, Z^{N\times N}) - D^{\mathrm{PH}}_{0,m}(x^{N\times N}, Z^{N\times N}) + D^{\mathrm{PH}}_{0,m}(x^{N\times N}, Z^{N\times N}) - D^{2\mathrm{D}}_{0,m}(x^{N\times N}, Z^{N\times N}) > \epsilon\Big)\\ &\le \Pr\Big(L_{\hat{\mathbf{X}}^{n,0,m}_{2\mathrm{D\ univ}}}(x^{N\times N}, Z^{N\times N}) - D^{\mathrm{PH}}_{0,m}(x^{N\times N}, Z^{N\times N}) > \epsilon\Big),\end{aligned} \tag{15}$$

since

$$\Pr\big(D^{\mathrm{PH}}_{0,m}(x^{N\times N}, Z^{N\times N}) - D^{2\mathrm{D}}_{0,m}(x^{N\times N}, Z^{N\times N}) > 0\big) = 0$$

from Lemma 2. Therefore, the event in (15) becomes identical to the one in the 1-D problem, and the probability bound (14) is obtained by the exactly same analysis given in [2, Theorem 2]. ■

3.4.2 Competing with Combinations of k-th order denoisers

Establishing the result on the single-symbol denoiser case, we now can move on to the general case of competing with k-th order denoisers. For general $k > 0$, let $\tilde{k} = \lceil k/4\rceil$ and $n_k = (N-\tilde{k})\times(N-\tilde{k})$. Define

$$\mathcal{T}_{N_k} \triangleq \{t \in \mathbb{Z}^2 : t = (t_1, t_2), \tilde{k}+1 \le t_1 \le N-\tilde{k}, \tilde{k}+1 \le t_2 \le N-\tilde{k}\},$$

which is a subset of $\mathcal{T}_N$ with size $|\mathcal{T}_{N_k}| = n_k$. For given $z^{N\times N}$, we define n_k-tuple of (k-th order denoiser induced) single-symbol denoisers

$$\mathbf{S}_k(z^{N\times N}) \triangleq \{s_{k,t}(\mathbf{c}_{t,2\mathrm{D}}, \cdot)\}_{t\in\mathcal{T}_{N_k}} \in \mathcal{S}_0^{n_k},$$

where $\mathbf{c}_{t,2\mathrm{D}}$ is the 2-D k-th order context defined in Section 3.1. For brevity, we suppress the dependence on $z^{N\times N}$ in $\mathbf{S}_k(z^{N\times N})$ and denote it as $\mathbf{S}_k$. Similarly as in the case of $k = 0$, for given $\mathbf{S}_k$, we can associate the 2-D n-block denoiser $\hat{\mathbf{X}}^{n,\mathbf{S}_k}_{2\mathrm{D}}$ as

$$\hat{\mathbf{X}}^{\mathbf{S}_k}_{t,2\mathrm{D}}(z^{N\times N}) = s_{k,t}(\mathbf{c}_{t,2\mathrm{D}}, z_t) \quad t \in \mathcal{T}_{N_k}.$$

As in the 1-D case [2], for each $\mathbf{c} \in \mathbf{C}_{k,2\mathrm{D}}$, we define $\mathcal{T}(\mathbf{c}) = \{\tau : \mathbf{c}_{\tau,2\mathrm{D}} = \mathbf{c}, \tau \in \mathcal{T}_{N_k}\}$ as the set of indices of $z^{N\times N}$ where the 2-D context equals $\mathbf{c}$. Then, for fixed m, each quadtree $q \in Q_m$, and given $z^{N\times N}$, define a subset $\mathbf{S}_{k,m}(q) \subset \mathcal{S}_0^{n_k}$ as

$$\mathbf{S}_{k,m}(q) \triangleq \bigcup_{\mathbf{c}\in\mathbf{C}_{k,2\mathrm{D}}} \big\{\{s_{k,t}(\mathbf{c},\cdot)\}_{t\in\mathcal{T}(\mathbf{c})} : s_{k,i}(\mathbf{c},\cdot) = s_{k,j}(\mathbf{c},\cdot) \text{ if } R_q(i) = R_q(j)\big\},$$

the set of n_k-tuples of (k-th order denoiser induced) single-symbol denoisers that, within the sub-data points $\{t : t \in \mathcal{T}(\mathbf{c})\}$ for each $\mathbf{c} \in \mathbf{C}_{k,2\mathrm{D}}$, are identical on the regions decomposed by $q \in Q_m$. Then, define

$$\mathcal{S}^n_{k,Q_m}(z^{N\times N}) = \bigcup_{q\in Q_m} \mathbf{S}_{k,m}(q)$$

as all combinations of (k-th order denoiser induced) single-symbol denoisers that, within each subsequence associated with each $\mathbf{c} \in \mathbf{C}_{k,2\mathrm{D}}$, is confined to remain constant within each of the regions determined by QTs with m leaves. Again, for brevity, the dependence on $z^{N\times N}$ in $\mathcal{S}^n_{k,Q_m}(z^{N\times N})$ is suppressed, and we write $\mathcal{S}^n_{k,Q_m}$. Note that the

above notation simply generalizes that of Section 3.4.1 by parallelizing over each 2-D context. Finally, in analogy to (10), for given $(x^{N\times N}, z^{N\times N})$, we define the k-th order performance target as

$$D^{\rm 2D}_{k,m}(x^{N\times N}, z^{N\times N}) = \min_{\mathbf{S}\in\mathcal{S}^n_{k,Q_m}} \frac{1}{n_k}\sum_{t\in\mathcal{T}_{N_k}} \Lambda(x_t, s_{k,t}(\mathbf{c}_{t,\rm 2D}, z_t)). \tag{16}$$

Here too, using the estimated loss to directly find the combination of k-th order 2-D sliding window denoisers in $\mathcal{S}^n_{k,Q_m}$ that achieves (16) may require prohibitive complexity in m. Therefore, as in Section 3.4.1, we consider the 2-D data in the order of PH scanning, this time independently for each subsequence defined by each 2-D context $\mathbf{c}\in\mathbf{C}_{k,\rm 2D}$. To that end, we define $\mathcal{S}^{{\rm PH}(n)}_{k,m}$ that parallels (12) and [2, (28)] as

$$\mathcal{S}^{{\rm PH}(n)}_{k,m} = \{\mathbf{S}_k : \{s_{k,\tau}(\mathbf{c},\cdot)\}_{\tau\in\mathcal{T}(\mathbf{c})} \in \mathcal{S}^{{\rm PH}(n(\mathbf{c}))}_{0,m(\mathbf{c})} \ \text{ for all } \ \mathbf{c}\in\mathbf{C}_{k,\rm 2D}\}, \tag{17}$$

where $n(\mathbf{c}) = |\mathcal{T}(\mathbf{c})|$ and $m(\mathbf{c}) = \min\{m, n(\mathbf{c})\}$. Although it may look complex, (17) is simply a set of combination of k-th order sliding window denoisers that shift at most m times along the PH-scanned subsequence for each 2-D context $\mathbf{c}\in\mathbf{C}_{k,\rm 2D}$. With this notation and definitions, we define

$$\hat{\mathbf{S}}_{k,m} = \arg\min_{\mathbf{S}_k\in\mathcal{S}^{{\rm PH}(n)}_{k,m}} \frac{1}{n_k}\sum_{\{i:{\rm PH}_i\in\mathcal{T}_{N_k}\}} \ell(Z_{{\rm PH}_i}, s_{k,{\rm PH}_i}(\mathbf{c}_{{\rm PH}_i,\rm 2D},\cdot)) \tag{18}$$

and the 2-D (k,m)-S-DUDE, $\hat{\mathbf{X}}^{n,k,m}_{\text{2-D univ}}$, as $\hat{\mathbf{X}}^{n,\hat{\mathbf{S}}_{k,m}}_{\rm 2D}$. By applying the dynamic programming algorithm of the 1-D case [2] for each PH-scanned subsequence defined by each context $\mathbf{c}\in\mathbf{C}_{k,\rm 2D}$, we can find (18) with complexity linear in both n and m. A subtle point to emphasize here is that although we apply the 1-D scheme on the PH scanned 1-D data, the subsequences that we apply our algorithm on are defined by the 2-D k-th order contexts. In this way, our scheme still competes with all combinations of 2-D k-th order sliding window denoisers with high probability, as is established in the following result.

Theorem 3 *For all $\epsilon>0$ and $x^{N\times N}\in\mathcal{X}^{N\times N}$,*

$$\begin{aligned}&\Pr\Big(L_{\hat{\mathbf{X}}^{n,k,m}_{\text{2-D univ}}}(x^{N\times N}, Z^{N\times N}) - D^{2D}_{k,m}(x^{N\times N}, Z^{N\times N}) > \epsilon\Big)\\ \le\ & 2(\tilde{k}+1)^2\exp\Big(-n_k\cdot\Big[\frac{(\epsilon/L_{\max})^2}{2(\tilde{k}+1)^2} - 2|\mathcal{Z}|^{2k}\cdot\Big\{h\Big(\frac{m}{n_k}\Big) + \frac{(m+1)\ln|\mathcal{S}|}{n_k}\Big\}\Big]\Big).\end{aligned} \tag{19}$$

Proof : Once we define

$$D^{\rm PH}_{k,m}(x^{N\times N}, z^{N\times N}) = \min_{\mathbf{S}_k\in\mathcal{S}^{{\rm PH}(n)}_{k,m}} \frac{1}{n_k}\sum_{\{i:{\rm PH}_i\in\mathcal{T}_{N_k}\}} \Lambda(Z_{{\rm PH}_i}, s_{k,{\rm PH}_i}(\mathbf{c}_{{\rm PH}_i,\rm 2D},\cdot)),$$

we can again easily see that

$$D^{\rm PH}_{k,m}(x^{N\times N}, z^{N\times N}) \le D^{\rm 2D}_{k,m}(x^{N\times N}, z^{N\times N})$$

for all $(x^{N\times N}, z^{N\times N})$ since $\mathcal{S}^{{\rm PH}(n)}_{k,m}$ is a larger set than $\mathcal{S}^n_{k,Q_m}$. Hence, proving the theorem becomes showing

$$\Pr\Big(L_{\hat{\mathbf{X}}^{n,k,m}_{\text{2-D univ}}}(x^{N\times N}, Z^{N\times N}) - D_{k,m}(x^n_{\rm PH}, Z^n_{\rm PH}) > \epsilon\Big) \le (19),$$

which can be derived by the identical argument as in [2, Theorem 3]. ■

The following result, which is a direct consequence of the above theorem, can be considered the analogue of Theorem 1(a) to 2-D data. It shows that our algorithm is still universal, i.e., regardless of the underlying data, our algorithm asymptotically attains the optimum performance in the reference class.

Theorem 4 *Suppose $k=k_n$ and $m=m_n$ are such that (19) is summable in n, e.g., $k=c_1\log n$ with $c_1<\frac{1}{2\log|\mathcal{Z}|}$ and $m=n^\alpha$ with $\alpha<1$. Then, for all $\mathbf{x}\in\mathcal{X}^{\infty\times\infty}$, the sequence of denoisers $\{\hat{\mathbf{X}}^{n,k,m}_{\text{2D univ}}\}$ satisfies*

$$\lim_{N\to\infty}\big[L_{\hat{\mathbf{X}}^{n,k,m}_{\text{2D univ}}}(x^{N\times N}, Z^{N\times N}) - D^{2D}_{k,m}(x^{N\times N}, Z^{N\times N})\big] = 0 \ \text{ a.s.}$$

Proof : The proof combines the summability condition, the Borel-Cantelli lemma, the bound (19), and simple use of the union bound, similarly as in the proof of Theorem 1 in [2]. ■

3.5 Algorithm and complexity

We have shown that by applying the 1-D S-DUDE algorithm in [2, Section V-A] separately on the PH scanned subsequences for each 2-D context $\mathbf{c} \in \mathbf{C}_k$, the resulting scheme can attain the performance of the best combination of the k-th order denoisers that shifts across the m separate quadtree decomposed regions. The pseudo-algorithm for our 2-D S-DUDE is given below:

Algorithm 1 The two-dimensional (2-D) (k, m)-Shifting DUDE

Require: $LM_t \in \mathbb{R}^{(m+1)\times|\mathcal{S}|}$, $IM_t \in \mathbb{R}^{|\mathcal{S}|}$ for $t \in \mathcal{T}_k$, $T \in \mathbb{R}^{|\mathbf{C}_k|}, r \in \mathbb{R}^{|\mathbf{C}_k|}, q \in \mathbb{R}^{|\mathbf{C}_k|}, L \in \mathbb{R}$ as in [2, Section V-A]
Ensure: $\hat{\mathbf{S}}_{k,m} = \{\hat{s}_{k,t}(\mathbf{c}_{t,2\mathrm{D}}, \cdot)\}_{t\in\mathcal{T}_k}$ in (18) and the denoised output $\{\hat{x}_t\}_{t\in\mathcal{T}_k}$
  for increasing order of PH scanned index $\{i : \mathrm{PH}_i \in \mathcal{T}_k\}$ **do**
    identify the 2-D context $\mathbf{c}_{\mathrm{PH}_i,2\mathrm{D}}$
    run the forward recursion of 1D S-DUDE on PH scanned points $\{\mathrm{PH}_j : \mathrm{PH}_j \in \mathcal{T}(\mathbf{c}_{\mathrm{PH}_i,2\mathrm{D}}), \quad j \le i\}$
  end for
  for decreasing order of PH scanned index $\{i : \mathrm{PH}_i \in \mathcal{T}_k\}$ **do**
    identify the 2-D context $\mathbf{c}_{\mathrm{PH}_i,2\mathrm{D}}$
    run the backward recursion of 1D S-DUDE on PH scanned points $\{\mathrm{PH}_j : \mathrm{PH}_j \in \mathcal{T}(\mathbf{c}_{\mathrm{PH}_i,2\mathrm{D}}), \quad j \ge i\}$
    identify the best single-symbol denoiser $\hat{s}_t(\mathbf{c}_{\mathrm{PH},2\mathrm{D}}, \cdot)$ for location PH_i
    obtain the denoised symbol $\hat{x}_{\mathrm{PH}_i} = \hat{s}_t(\mathbf{c}_{\mathrm{PH},2\mathrm{D}}, z_{\mathrm{PH}_i})$
  end for

Note that the PH scanning of the data and running the 1-D S-DUDE on those scanned points can be done simultaneously, not separately. Therefore, the time and memory complexities of our algorithm are exactly the same as those of 1-D S-DUDE : $O(nm)$. Hence, competing with the QT decomposed, shifting 2-D sliding window denoisers for 2-D data is no harder than competing with shifting sliding window denoisers for 1-D data.

3.6 Remarks

Before presenting the experimental results of our scheme, we have a few remarks regarding our algorithm and analyses in above subsections.

1. The performance guarantee results on expected (rather than actual) loss, and on a stochastic setting where the noise-free image, rather than an individual data array, is a random field, can be derived similarly as in the settings of [1] and [2], once equipped with the above semi-stochastic setting result in Theorem 4. We omit the exercise for conciseness and to refrain from repetition.

2. It may also be natural to conceive of a denoising algorithm that heuristically finds a QT decomposition by greedily merging child nodes as in [6], using the estimated loss. This scheme is practical, and may be competitive with the best shifting sliding window denoisers based on QT decomposition, but is difficult to analyze and obtain rigorous performance guarantees. On the other hand, as the results above guarantee, our scheme, achieves the best possible performance among all scheme in the same reference class, and is practically implementable.

3. The two components of our scheme, namely, QT decomposition and PH scanning, have been developed independently in previous literature, but in the denoising setting, we see that the marriage of the two is natural since they play complementary roles; the former efficiently segments data points that have similar characteristics and the latter unfolds the 2-D data into 1-D data while preserving the local similarity attained by the former.

4. Although our algorithm and analysis pertained exclusively to 2-D data case, it is not hard to extend our scheme to the multi-dimensional data case beyond 2-D case, since analogously defining QT decomposition and PH scanning for multi-dimensional data is straightforward.

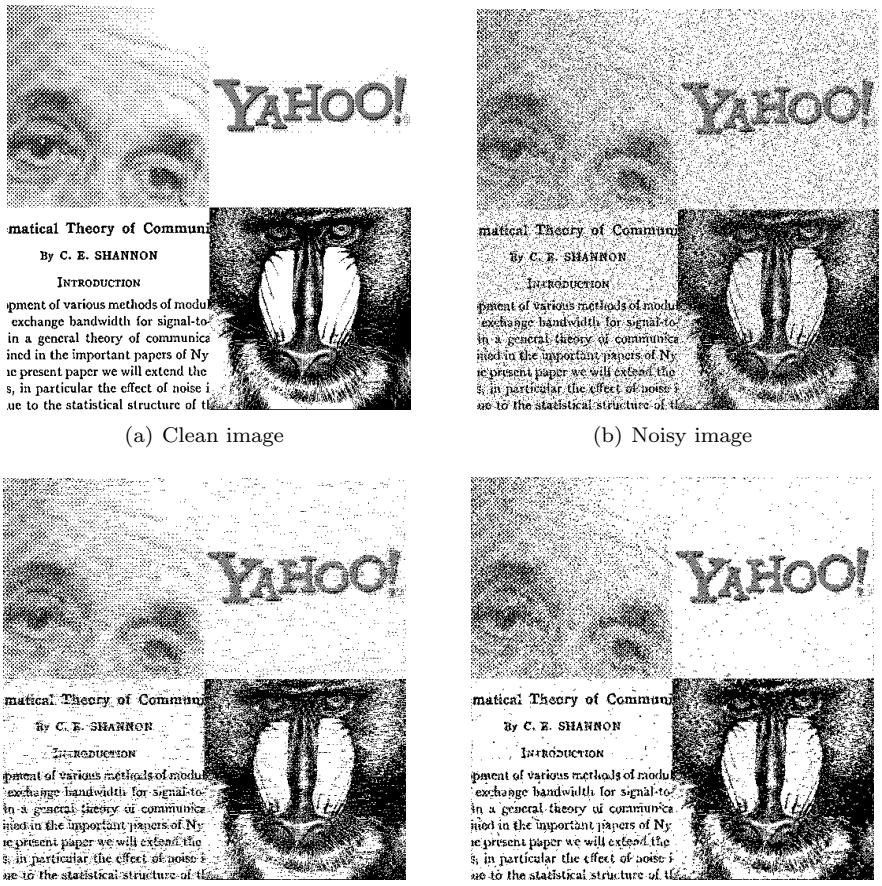
4 Experimental Results

As shown above, our 2-D S-DUDE enjoys considerable performance guarantees and efficient implementation, however, it is not clear how effective it might be in practice to compete with the reference class of QT decomposition-based

shifting sliding window denoisers. Hence, we show the performance of our scheme on three different sample images and compare with baseline schemes to highlight when use of our algorithm could be advantageous.

4.1 Synthetic test image

First, we show the denoising results for a synthetic image that showcase the benefit of our 2-D S-DUDE scheme. Figure 6(a) shows the clean image that was constructed by pasting four binary sub-images that have different characteristics, and Figure 6(b) is the noisy counterpart corrupted by a binary symmetric channel (BSC) with crossover probability $\delta = 0.1$. The total image size is 512×512 and each sub-image has size 256×256. We compare our scheme with



(a) Clean image (b) Noisy image

(c) Raster scan+1D S-DUDE ($m = 4$) (d) 2D S-DUDE ($m = 4$)

Figure 6: Synthetic test image and denoising results

three different baselines;

(i) **2-D DUDE** : This simply generalizes the 1-D DUDE in [1] to the 2-D data case with using the 2-D contexts defined in Section 3.1. Note that this scheme is already superior to many state-of-the-art image denoising schemes, as reported in [1, Section VIII-C] and [3]. It has a single parameter k, the context size for the 2-D context.

(ii) **1-D S-DUDE after raster scanning the data**: This scheme first does the simple horizontal raster scan of the image, then applies the 1-D S-DUDE developed in [2] on the resulting 1-D sequence. It is the scheme used in [2, Section V-A] for image denoising experiments. The scheme has two parameters - k for the context size for the 1-D context and m for the number of shifts.

(iii) **1-D DUDE after raster scanning the data**: This scheme is a baseline, which coincides with the scheme in (ii) when the number of shifts m is set to 0. In [2, Section V-A], (ii) was shown to be superior to this scheme for images with characteristics that are abruptly changing.

One may think that the only difference between our 2-D S-DUDE and scheme (ii) is that we use the PH scan in place of the raster scan, but there is also a subtle difference that we consider the subsequence points with respect to the 2-D contexts whereas (ii) simply considers the 1-D contexts from the raster-scanned 1-D sequence.

Figure 7 shows the bit error rate (BER) results for the four schemes. For (ii) and 2-D S-DUDE, we only show the results for the best m value, which happened to be $m = 4$ for both schemes. First, we can see that the difference between (ii) and (iii) is small. This is well expected since we can easily notice that the characteristics of the raster-scanned 1-D sequence of Figure 6(a) vary linearly with respect to the sequence length n, which violates the necessary condition on the number of shifts m for (ii) to work, i.e., m should be sublinear in n as specified in [2, Theorem 5]. Second, interestingly, there is also no big difference between (i), which uses the 2-D context, and (ii) and (iii), which use the 1-D context from the raster-scanned sequence. Probably this shows that for our image, the 1-D contexts from the raster-scanned sequence are enough for capturing the locality of images, but we do not know whether this would be a general phenomenon. Finally, we can observe that our 2-D S-DUDE with $m = 4$ clearly dominates all three

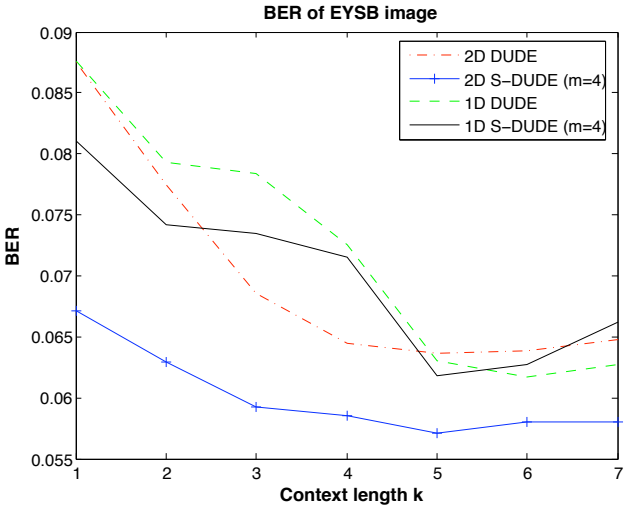


Figure 7: Bit error rate comparison for the synthetic image

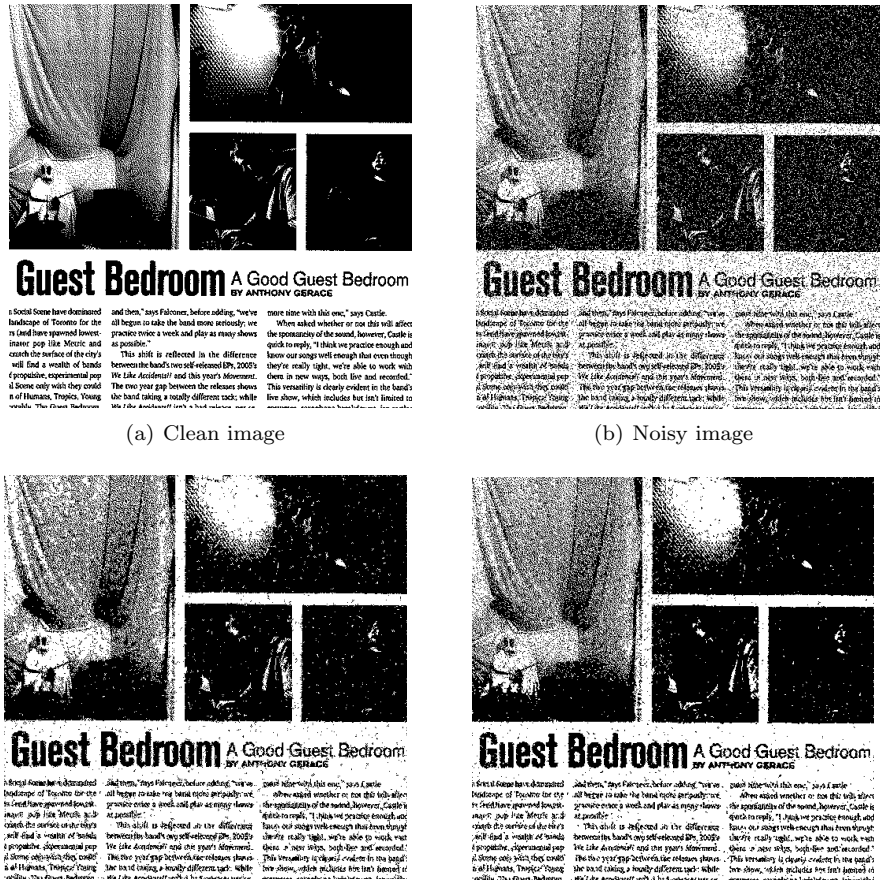
baseline schemes. Note that, by construction, it would be optimal to first decompose the image in Figure 6(b) into four separate quadrants (quadtree with 4 leaves) and apply four independent denoisers in each region. We see that by considering the noisy pixels in the order of PH scanning, our scheme, which knows nothing about the underlying clean image, successfully learns the decomposition of the image and further reduces the BER for denoising compared to other baseline schemes.

The resulting denoised images for scheme (ii) and our 2-D S-DUDE are shown in Figure 6(c) and Figure 6(d), respectively. In line with the BER plot in Figure 7, we visually see that 2-D S-DUDE is superior to the scheme (ii) not only in terms of the number of errors, but also in terms of detecting the boundaries of images and preserving the textures. Particularly, the texts in Figure 6(d) are more readable and the boundary between the Einstein and Yahoo! images is more clearly captured in Figure 6(d).

4.2 Scanned magazine image

Although the result on the synthetic image is encouraging, one may suspect the image was constructed in favor of 2-D S-DUDE, since it was divided into different sub-images corresponding to QT quadrants. We thus test our algorithm on a real image. Figure 8(a) shows the clean binary image obtained from scanning a real magazine page. Unlike the synthetic image in Section 4.1, this image represents the common and realistic characteristics of images that have different textures in different regions of images. The image size is 512×512, and again corrupted by BSC with $\delta = 0.1$ that led to the noisy image in Figure 8(b).

Here, we only compare our 2-D S-DUDE with 2-D DUDE, scheme (i) in the previous subsection, as there were no significant differences between other baselines in Section 4.1, and it is more natural to compare our scheme with



(a) Clean image (b) Noisy image

(c) 2D DUDE ($k = 8$) (d) 2D S-DUDE ($k = 4, m = 4$)

Figure 8: Scanned magazine image and denoising results

one that also uses the 2-D contexts. For our experiments, we varied the context size k from 1 to 9 for both of the schemes and tried several values of m for 2-D S-DUDE. The BER plot in Figure 9 again shows that our 2-D S-DUDE consistently outperforms 2-D DUDE and the best BER is reduced by about 6%. This improvement is significant since 2-D DUDE was already shown to outperform many of the state-of-the-art binary image denoising algorithms. Moreover, we see that the 2-D S-DUDE achieves its optimum performance using context length k which is half of that used by the 2-D DUDE, resulting in overall lower complexity. That is, although 2-D S-DUDE introduces another parameter m, the complexity is linear in m, and it reduces the dependency on k which contributes exponentially to the complexity. Figure 8(c) and Figure 8(d) respectively show the denoising results. We observe that the 2-D S-DUDE not only has a smaller number of errors, but also does a better job than 2-D DUDE in preserving the sub-image textures.

4.3 Lena image

We now show the results for the binary Lena image of size 512×512. The noisy channel is identical to the previous subsections. Figure 10(a) and Figure 10(b) show the clean and noisy Lena images, and Figure 10(c) shows the denoising results for both 2-D DUDE with $k = 6$ and 2-D S-DUDE with $k = 6$ and $m = 4$. Figure 10(d) shows the BER plot for both 2-D DUDE and 2-D S-DUDE.

These results show what might be expected; when the image characteristics are largely homogeneous, the best denoising performance of 2-D DUDE and 2-D S-DUDE are similar. However, as we can see in the BER plot in Figure 10(d), 2-D S-DUDE reduces the BER faster than 2-D DUDE even with smaller context size k by introducing another parameter m. This can be beneficial when we do not know which k would be the optimal for denoising in a practical scenario.

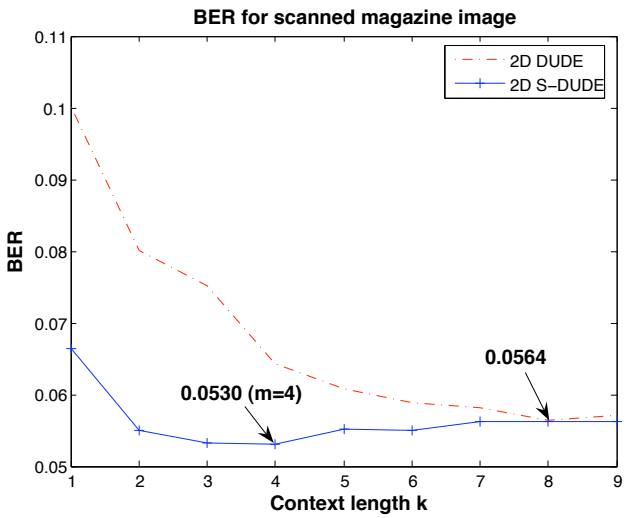


Figure 9: Bit error rate comparison for the scanned magazine image

5 Concluding Remarks

We have generalized the S-DUDE proposed in [2] to two-dimensional data. Due to the hardness of optimally segmenting the 2-D data, we introduced a QT decomposition-based reference class of shifting 2-D sliding window denoisers, then utilized the PH scanning technique to efficiently implement the scheme that can attain the optimum performance in the reference class without knowing anything a priori about the characteristics of the underlying clean data. Experimental results show that our scheme can be effective in further reducing the loss of 2-D DUDE, especially for heterogenous images consisting of sub-images of varying natures. Among other related lines of inquiry, future work will investigate the effectiveness of combining more general data segmentation and scanning techniques.

Acknowledgement

The authors are grateful to Erik Ordentlich for helpful discussions.

References

[1] T. Weissman, E. Ordentlich, G. Seroussi, S. Verdú, and M. Weinberger, "Universal discrete denoising: Known channel," *IEEE Trans. Inform. Theory*, vol. 51, no. 1, pp. 5–28, 2005.

[2] T. Moon and T. Weissman, "Discrete denoising with shifts," *IEEE Trans. Inform. Theory*, vol. 55, no. 11, pp. 5284–5301, 2009.

[3] E. Ordentlich, G. Seroussi, S. Verdú, M. Weinberger, and T. Weissman, "A discrete universal denoiser and its application to binary images," *Proc. IEEE Int. Conf. on Image Processing*, p. 117-120, vol. 1, Sept. 2003.

[4] A. Lempel and J. Ziv, "Compression of two-dimensional data," *IEEE Trans. Inform. Theory*, vol. 32, no. 1, pp. 2–8, 1986.

[5] H. Sagan, *Space-Filling Curves*. Springer-Verlag, 1994.

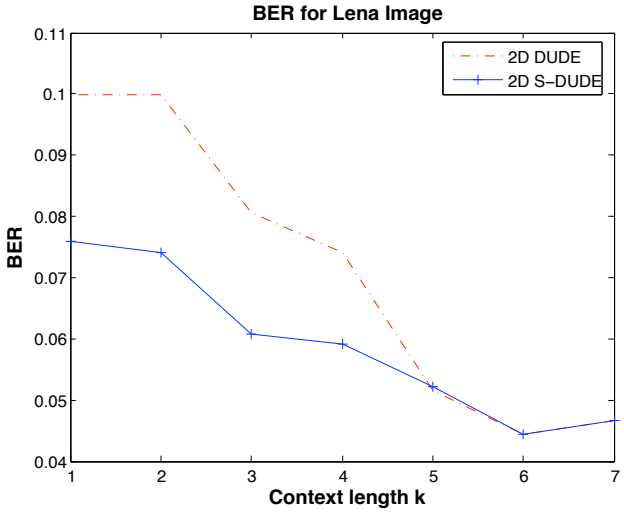
[6] E. Shusterman and M. Feder, "Image compression via improved quadtree decomposition algorithms," *IEEE Trans. Image Processing*, vol. 3, no. 2, pp. 207–215, 1994.

[7] G. Sullivan and R. Baker, "Efficient quadtree coding of images and video," *IEEE Trans. Image Processing*, vol. 3, no. 3, pp. 327–331, 1994.

(a) Clean image

(b) Noisy image

(c) 2D DUDE($k = 6$) and 2D S-DUDE ($k = 6, m = 4$)



(d) BER plot

Figure 10: Lena image and denoising results

[8] J. Smith and S.-F. Chang, "Quad-tree segmentation for texture-based image query," *Proceedings of the second ACM international conference on Multimedia*, pp. 279–286, 1994.

[9] R. Shukla and M. Vetterli, "Geometrical image denoising using quadtree segmentation," *IEEE Int. Conf. on Image Processing*, vol. 2, pp. 1213–1216, 2004.

[10] T. Weissman and S. Mannor, "Universal compression of multi- dimensional data arrays using self-similar curves," *Proc. 38th Annu. Allerton Conf. Communication, Control, and Computing*, pp.470-479, October 2000.

[11] A. Cohen, T. Weissman, and N. Merhav, "Scanning and sequential decision making for multi-dimensional data - Part I: the Noiseless case," *IEEE Trans. Inform. Theory*, vol. 53, no. 9, pp. 3001–3020, 2007.

[12] A. Cohen, T. Weissman, and N. Merhav, "Universal scanning and sequential decision making for multi-dimensional data, Part II: the Noisy case," *IEEE Trans. Inform. Theory*, vol. 54, no. 12, pp. 5609–5631, 2008.

[13] T. Weissman, E. Ordentlich, M. Weinberger, A. Somekh-Baruch, and N. Merhav, "Universal filtering via prediction," *IEEE Trans. Inform. Theory*, vol. 53, no. 4, pp. 1253–1264, 2007.