A Near Optimal Coder For Image Geometry With Adaptive Partitioning

Arian Maleki[†] , arianm@stanford.edu
Morteza Shahram[‡] , mshahram@stanford.edu
Gunnar Carlsson [*] , gunnar@math.stanford.edu

Abstract

In this paper, we present a new framework to compress the geometry of images. This framework generalizes the standard quad partitioning approaches in compression of image geometry (e.g. wedgelet) in two ways. First, we employ an adaptive rectangular partitioning rather than quadratic partitioning. Second, our coder uses an overcomplete collection of (stripe-like) atoms which contains wedgelets as a special case. We present an information-theoretical analysis based on Kolmogorov's ϵ-entropy to show that this collection provides a near-optimal representation of a class of cartoon images with piecewise polynomial boundaries. Furthermore, we develop a provably near-optimal greedy algorithm that significantly reduces the complexity of the exhaustive search method required to achieve the entropy bound. Simulation results for the rate distortion shows a 1.5-2 dB improvement over the standard wedgelets for the "Cameraman" image.

I. Introduction

Natural images are composed of two components: cartoon (geometry) and texture. Cartoons are conveniently described by geometric structures such as flat objects with piecewise smooth boundaries. It is known that wavelets do not provide sparse representation for these structures [1]. Several works in the past have proposed frameworks to achieve more efficient representations [2], [3], [4], [5] of geometrical features such as lines and curves. Among these frameworks, wedgelets in particular have shown to be a successful base for practical compression schemes [6], [7]. The basic idea of wedgelets is to partition an image into dyadic squares and approximate each square with a wedge-like patch. However, wedgelets have emerged from optimal representations of a very simple class of images (bi-color patches with smooth boundary)[2], [8]. The work by Shukla et al. [11] implies that the wedgelets system loses coding efficiency (due to overpartitioning) for regions with slightly more complicated geometries which are frequent in natural images. Furthermore, there have been several works on learning a dictionary by which image patches can be represented very sparsely [9], [10]. In these approaches, it is assumed that there exists a dictionary of 2-D functions such that any image patch can be approximated as a sparse linear combination of the elements of the dictionary. The elements of such dictionary are learned by an iterative optimization scheme. The majority of learned elements resemble stripe-like patches rather than wedgelets.

In this paper, we introduce a new class of 2-D functions to deal with more general geometries than those of wedgelets. We study the performance of an optimal encoder for this class of functions by computing Kolmogorov's ϵ-entropy. Kolmogorov's ϵ-entropy [13] encodes a compact set of functions such that the distortion of all signals is less than ϵ and counts the number of the required codewords. Shannon's rate-distortion theory also provides a framework to study compression. It codes the stochastic sources by discarding the functions with small probabilities and compressing the rest. Since we do not assume any probabilistic model for our class of functions, we use Kolmogorov's ϵ-entropy.

In order to reach the ϵ-entropy we expand the dictionary of wedgelets with stripe-like patches (which we call bi-wedgelets) and also generalize the partitioning algorithm. Using exhaustive search for this algorithm

[†] Department of Electrical Enginerring, Stanford University
[‡] Department of Statistics, Stanford University
[*] Department of Mathematics, Stanford University

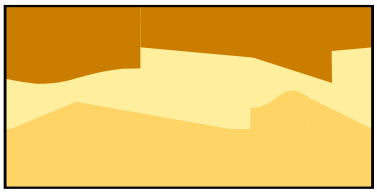

Fig. 1. A sample function in $\mathcal{T}_N^Q(I_x)$

is computationally infeasible. We propose a greedy partitioning and greedy atom selection methodology to significantly speed up the encoder.

The organization of this paper is as follows. In the next section, we introduce a class of functions which will be our model for image geometries throughout this paper. We also compute the ϵ-entropy of this class. In Section III, a practical near-optimal coder will be proposed. In Section IV, we present rate distortion results for different algorithms. Finally, we conclude the paper in Section V.

II. Notations and Framework

Let the space of piecewise polynomial functions on an interval $I_x \subset [0,1]$ be represented by $\mathcal{BP}_N^Q(I_x, A)$ where N is the maximum degree of the polynomials on the interval, Q is the maximum number of singularities[1], and the functions are bounded by A, i.e. $\sup_{t\in I_x} |f(t)| \leq A$. Based on the class of $\mathcal{BP}_N^Q(I_x, A)$, we construct a class of 2-D functions, $\mathcal{T}_N^Q(I_x)$, whose elements are defined by

$$f(x,y) = \alpha_d \mathbb{1}_{\{y\leq h_d(x)\}} + \alpha_m \mathbb{1}_{\{h_d(x)<y\leq h_u(x)\}} + \alpha_u \mathbb{1}_{\{y>h_u(x)\}}, \tag{1}$$

where $h_u(x), h_d(x) \in \mathcal{BP}_N^Q(I_x, 1)$, $0 \leq h_d(x) \leq h_u(x)$, $\mathbb{1}_A$ is the indicator function of set A and $0 \leq \alpha_d, \alpha_m, \alpha_u \leq 1$. Figure 1 shows one of the functions belonging to this space[2]. We regard this class as a subset of $L_2([0,1]^2)$, and our distortion measure is the mean square error, i.e.,square of L_2-norm. This class will serve as our model for cartoon images. The goal is to find some bounds on the performance of the optimal compression scheme. The best performance is given by Kolmogorov's ϵ-entropy defined as

$$H_\epsilon(\mathcal{T}_N^Q(I_x)) = \log_2 N_\epsilon(\mathcal{T}_N^Q(I_x)), \tag{2}$$

where $N_\epsilon(\mathcal{T}_N^Q(I_x))$ is the minimum number of sets in an ϵ-covering of the set $\mathcal{T}_N^Q(I_x)$ [13],[14]. For coding an element of $\mathcal{T}_N^Q(I_x)$ with the distortion at most ϵ the best achievable rate is $H_\epsilon(\mathcal{T}_N^Q(I_x))$. The following theorem shows the performance bounds of an optimal encoder.

Theorem 2.1: There exist two positive constants D_1, D_2 such that the ϵ-entropy of the class $\mathcal{T}_N^Q(I_x)$ satisfies the following two constraints,

$$\begin{aligned} D_1 + (N+1)(Q+1)\log\left(\frac{1}{\epsilon}\right) &\leq H_\epsilon(\mathcal{T}_N^Q(I_x)) \\ &\leq D_2 + \frac{(4N+7)(Q+1)}{2}\log\left(\frac{1}{\epsilon}\right) \end{aligned} \tag{3}$$

where, D_1 and D_2 are two constants depending on N and Q.

Proof: See section VI. ■

In the next section, we will discuss a practical framework to overcome complexity of the exhaustive search associated with the ϵ-entropy.

[1]A singularity is defined as a point at which the function is not infinitely differentiable.

[2]This class does not consider the orientation which is present in images, but the algorithm we will propose in the next section is able to deal with non-horizontal orientations as well.

III. ALGORITHM

Kolmogorov's ϵ-entropy of $\mathcal{T}_N^Q(I_x)$ is found via an exhaustive search through a very large set of patches and therefore does not offer any practical scheme. In this section we propose a new algorithm which is computationally feasible and theoretically as successful as exhaustive search. It will be shown in Section IV that this algorithm outperforms wedgelets in coding natural images. The proposed coder partitions an image into small (in general rectangular) patches and then approximates each patch with only one element from the proposed set. A greedy partitioning is used to find the singularities. We then represent the partitioned patches with elements in $\mathcal{T}_N^0(I_x)$ also in a greedy fashion. For the sake of clarity, we present the algorithm for the horizontal case. However it is readily applicable to the general case.

A. Greedy Partitioning

Let $I = I_x \times I_y = [x_s, x_t] \times [y_s, y_t] \subset [0,1]^2$ be the interval in which the image patch g is defined. Our greedy algorithm partitions the interval I_x into $I_p = [x_s, p]$ and $I_p^c = [p, x_t]$ such that

$$(p^*, f_1^*, f_2^*) = \underset{p \in I_x, f_1 \in \mathcal{T}_N^0(I_p), f_2 \in \mathcal{T}_N^0(I_p^c)}{\arg\min} \|g - f_1 \mathbb{1}_{I_p} - f_2 \mathbb{1}_{I_p^c}\|_{L^2(I)}^2.$$

The following theorem shows the success of this greedy algorithm.

Theorem 3.1: The greedy partitioning algorithm is able to find the Q singularity points in at most $2Q+1$ steps. In other words, it will find $2Q+1$ partition points and Q of them are the actual singularity points.

Proof: See section VI. ■

One way to speed up the algorithm is to skip some of the pixels in the partitioning. The following theorem justifies the greedy algorithm in that case.

Theorem 3.2: Suppose that we only search for the partition points on a grid $\{0, \delta, 2\delta, \dots n\delta = 1\}$ and also assume that $\delta < \min_i(d_{i+1} - d_i)$. Then the greedy partitioning algorithm will find $3Q+1$ partition points and $2Q$ of these points are the neighboring points of the singularities.

Proof: The proof is very similar to the proof of Theorem 3.1. ■

Theorem 3.3: The rate-distortion performance of the greedy partitioning algorithm with the atoms selected from $\mathcal{T}_N^0$, satisfies

$$D(R) \leq c_1 2^{-\frac{R}{(4N+7)(Q+1)}} \tag{4}$$

in case of precise partitioning and

$$D(R) \leq c_2 2^{-\frac{2R}{3(4N+7)(Q+1)}} \tag{5}$$

in case of coarse partitioning, where c_1 and c_2 depend on N and Q.

Proof: See section VI. ■

By comparing the results of this theorem with the upper bound of the ϵ-entropy we see that the exponent in distortion-rate expression of our algorithm is different from the theoretical bound by a factor of two (or three in the case of coarse partitioning). The computationally expensive part of this algorithm is finding the best element of $\mathcal{T}_N^0$ which will be addressed in the next section.

B. Greedy Element Selection

Let each polynomial specifying $\mathcal{T}_N^0(I_x)$ be selected from a set with cardinality n. Therefore, for representing a given patch, finding the best element of $\mathcal{T}_N^0(I_x)$ by exhaustive search requires $O(n^2)$. However, this will be reduced to $O(n)$ by the following greedy algorithm. First, we define a new class of functions:

$$\mathcal{D}_N^Q(I_x) = \{f : f(x,y) = \beta_d \mathbb{1}_{\{y \leq h(x)\}} + \beta_u \mathbb{1}_{\{y > h(x)\}}\}, \tag{6}$$

where $h(x) \in \mathcal{BP}_N^Q(I_x, 1)$, $0 \leq h(x) < 1$ and $0 \leq \beta_d,\ \beta_u \leq 1$. We also define the operator T on elements of $\mathcal{D}_N^Q(I_x)$ which gives the characterizing polynomial $h(x)$. First, we look for the best matching element to g,

$$f_1^* = \underset{f_1 \in \mathcal{D}_N^0(I_x)}{\arg\min} \|g - f_1\|^2_{L^2(I)}.$$

Next, we search for a second element whose linear combination with f_1^* is the best approximation to g,

$$(\gamma^*, f_2^*) = \underset{\gamma \in \mathbb{R}, f_2 \in \mathcal{D}_N^0(I_x),\ T(f_1) \neq T(f_2)}{\arg\min} \|g - \gamma f_1 - f_2\|^2_{L^2(I)}$$

The following theorem shows optimality of this approach:

Theorem 3.4: For a given $g \in \mathcal{T}_N^Q(I_x)$, the algorithm above which greedily selects elements from $\mathcal{D}_N^Q(I_x)$, finds its characterizing parameters (h_u, h_d, α_u, α_m, α_d) in only two steps.

Proof: See section VI. ■

C. Summary of Compression Algorithm

In the context of image compression, the class $\mathcal{T}_1^0$ includes all the possible "horizontal" strip-like patches. For the purpose of implementation, we generalize this class to also cover the rotations of these stripes – we call this class $\mathcal{GT}_1^0$. This algorithm divides a given image into smaller patches and approximates each patch with a function from $\mathcal{GT}_1^0$ (we refer to this collection as "bi-wedgelets"). To partition an image, we use the greedy partitioning algorithm explained earlier in III-A in both horizontal and vertical directions. Successive application of this procedure generates a hierarchical (rectangular) partition that corresponds to a tree [15]. The deeper in the tree, the better approximation of the image is obtained but at the higher bit rate. In order to control the bit rate, we assign a cost function to each node of the tree as

$$\text{representation error} + \ \lambda \ \times \ \text{bit rate}, \tag{7}$$

where λ is a positive real number. To decide whether a patch should be partitioned or not, its cost function is compared with the total cost of its children. If greater, the patch will be partitioned and the partitioning point is added to the bit stream. Otherwise, the patch is not partitioned anymore and the parameters which specify the best approximating stripe are encoded instead. As pointed out in III-B, the greedy selection algorithm is computationally much more favorable for finding the best approximating stripe. Algorithm 1 outlines the proposed compression scheme.

The decoder follows similar steps that is partitioning points are read from the input bit stream and constructs the skeleton of the image partitions. Once a partition is made, its parameters are extracted, its corresponding bi-wedgelet is made, and superimposed on that rectangle.

IV. Simulation Results

In this section, we present the results of implementing three compression schemes: standard wedgelets with quad partitioning, wedgelets with greedy adaptive partitioning, and the proposed bi-wedgelets with greedy adaptive partitioning and greedy atom selection. We apply these methods to the 128×128-pixel image of Cameraman which contains both texture and geometry and obtain the rate distortion curves for these schemes (Figure 2). We see that greedy wedgelets and greedy bi-wedgelets yield 0.5-1.3 dB and 1.5-1.9 dB improvement over the classic wedgelets, repectively. These results imply that adaptive partitioning and expanding the collection both contribute to the performance.

Figure 3 depicts the reconstructed images (and their difference with the original image) obtained by these methods at approximately 0.21 bpp. Residuals show that the geometry is represented better by bi-wedgelets. We also note that the partitions generated by greedy partitioning is more adapted to the structure of the image. At higher bitrates, the adaptive bi-wedgelets show greater improvement, whereas

Algorithm 1 : Proposed Encoder

Require: Image patch g, λ
  $\Phi^0 \Leftarrow \{I\}$ (I: image domain)
  $j \Leftarrow 0$
  while $\Phi^j \neq \emptyset$ **do**
    $\Phi^{j+1} = \emptyset$
    while $\Phi^j \neq \emptyset$ **do**
      pick an element from Φ^j: I^j
      $\Phi^j = \Phi^j \backslash I^j$ ($\backslash$: set difference)
      Find the best stripe-like representation of g on I^j
      Find the best horizontal partitioning $\{I_L, I_R\}$
      Find the best vertical partitioning $\{I_U, I_D\}$
      Calculate bit budget and mean squared error for the above 3 steps, E, E_H, E_R and R, R_H, R_V
      if $E + \lambda R \leq \min\left(E_H + \lambda R_H, E_V + \lambda R_V\right)$ **then**
        Encode parameters of the approximating patch
      else
        if $E_H + \lambda R_H \leq E_V + \lambda R_V$ **then**
          $\Phi^{j+1} = \Phi^{j+1} \cup \{I_L\} \cup \{I_R\}$, encode the partition point
        else
          $\Phi^{j+1} = \Phi^{j+1} \cup \{I_U\} \cup \{I_D\}$, encode the partition point
        end if
      end if
    end while
    $j \Leftarrow j + 1$
  end while
  return Bit Stream

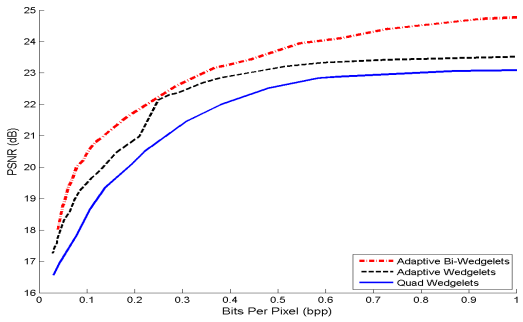

Fig. 2. Rate-distortion curves for the three codecs implemented in this paper.

the performance of adaptive wedgelets starts to saturate as well as that of quad wedgelets. This is due to the fact that at finer partitions, bi-wedgelets are more capable of capturing the geometry.

We wish to refer to Table I for complexity analysis of algorithms based on the choice of the partitioning method (quad or adaptive), collection of atoms (wedgelets or bi-wedglets), and method of finding best approximating atom (exhaustive or greedy). For instance, bi-wedgelets with adaptive partitioning have a $O(n^{3/2})$ flops complexity, where n represents the number of pixels.

V. Conclusion

In this paper we put forward an image coding framework, attempting to adapt to the geometry of natural images. We modeled images with varying-size partitions of stripes whose boundaries are considered to

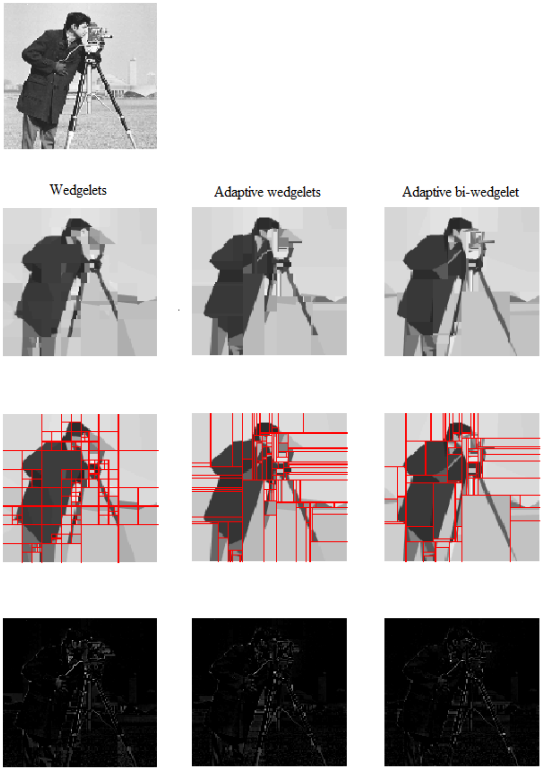

Fig. 3. Compression results for the encoders based on quad wedgelets (0.221 bpp, PSNR=20.50 dB), adaptive wedgelets (0.21 bpp, PSNR=20.97 dB), and adaptive bi-wedgelets (0.218 bpp, PSNR=21.94). Second, third and fourth rows show the reconstructed image, the partitions generated by each encoder, and the residuals.

TABLE I
COMPUTATIONAL COMPLEXITY OF CODERS DISCUSSED IN THIS PAPER.

Dictionary	Partitioning	Search	flops
Wedgelets	quadratic	-	$O(n)$
Wedgelets	adaptive greedy	-	$O(n^{3/2})$
Bi-Wedgelets	adaptive greedy	exhaustive	$O(n^{5/2})$
Bi-Wedgelets	adaptive greedy	greedy	$O(n^{3/2})$

be piecewise polynomials. We computed Kolmogorov's ϵ-entropy to show that our algorithm is a near-optimal coder in terms of the exponent of the distortion-rate. For implementation purposes, we worked with stripes with straight-line edges. In addition, we allowed the image to be divided into rectangular partitions which are adapted to the structure of the image.

There are several directions to pursue further. Theoretically, a more sophisticated analysis may lead to tighter bounds for the entropy expression. Also, the theory presented in this article to prove the optimality of greedy methods deals with the "ideal" case, where underlying patches are assumed to be stripes with piecewise polynomial boundaries. Therefore a very important issue is to develop stability results for the "non-ideal" scenarios. In terms of algorithm, we feel that a more sophisticated partitioning can improve the coding efficiency. Developing faster algorithms for element selection is also a promising task.

ACKNOWLEDGEMENT

This work was supported in part by DARPA grant HR0011-05-1-0007. We thank David Donoho for several insightful discussions.

REFERENCES

[1] M. Vetterli, "Wavelets, Approximation and Compression", IEEE Signal Processing Magazine, pp. 59-73, Sept. 2001.

[2] D. L. Donoho, "Wedgelets: Nearly minimax estimation of edges," *Annals of Stat.*, vol. 27, pp. 859-897, 1999.
[3] E. J. Candes, D. L. Donoho, "Curvelets–A surprisingly effective nonadaptive representation for objects with edges," in *Curves and Sureface fitting*, A. Cohen, C. Rabut and L. L. Shumaker, Eds. Nashvile, TN: Vanderbilt Univ. Press, 1999.
[4] E. J. Candes, D. L. Donoho, "Ridgelets: A key to higher-dimensional intermittency,"*Phil. Trans. R. Soc. Lond. A.*, pp. 2495-2509, 1999.
[5] M. N. Do, M. Vetterli, "The contourlet transform: an efficient directional multilresolution image representation," *IEEE Transactions on Image Processing*, vol. 14, Issue 12, pp. 2091- 2106, December 2005.
[6] M. B. Wakin, J. K. Romberg, H. Choi, R. G. Baraniuk , "Image compression using an efficient edge cartoon + texture model," IEEE Data Compression Conference, Utah, April 2002.
[7] M. B. Wakin, J. K. Romberg, H. Choi, and R. G. Baraniuk, "Rate-distortion optimized image compression using wedgelets," in IEEE Int. Conf. Image Process., vol. 3, Jun. 2002, pp. III-237III-240.
[8] M. N. Do, P. L. Dragotti, R. Shukla, M. Vetterli, "On the compression two-dimensional piecewise smooth functions," *IEEE International Conf. on Image Proc.*, Thessaloniki, Greece, Oct. 2001, pp. 14-17.
[9] B. A. Olshausen and D. J. Field, " Emergence of simple-cell receptive field properties by learning a sparse code for natural images," *Natrure*, 607-609, 1996.
[10] M. Aharon, M. Elad, A. M. Bruckstein, "K-SVD: an algorithm for designing overcomplete dictionaries for sparse representation," *IEEE Trans. on Signal Proc.*, Vol. 54, No. 11, pp. 4311-4322, Nov. 2006.
[11] R. Shukla, P. L. Dragotti, M. N. Do, M. Vetterli, "Rate-distortion optimized tree-structured compression algorithms for piecewise polynomial images," *IEEE Transactions on Image Processing*, vol. 14, no. 3, pp. 1-17, March 2005.
[12] T. M. Cover, J. A. Thomas, *Elements of Information Theory*, New York, Wiley, 1991.
[13] A. N. Kolmogorov, V. M. Tihomirov, "ϵ-entropy and ϵ-capacity of sets in functional spaces," *American Math. Society Translation* (Ser. 2), vol. 17, pp. 277-364, 1961.
[14] H. L. Royden, *Real Analysis*, 2nd Edition, New york:Macmillan, 1968.
[15] T. Hastie, R. Tibshirani, and J. Friedman, *The Elements of Statistical Learning*, Springer, 2001.

VI. PROOFS

Theorem 6.1: There exist two positive constants D_1, D_2 such that the ϵ-entropy of the class $\mathcal{T}_N^Q(I_x)$ satisfies the following two constraints,

$$\begin{aligned} D_1 + (N+1)(Q+1)\log\left(\frac{1}{\epsilon}\right) &\leq H_\epsilon(\mathcal{T}_N^Q(I_x)) \\ &\leq D_2 + \frac{(4N+7)(Q+1)}{2}\log\left(\frac{1}{\epsilon}\right) \end{aligned} \tag{8}$$

where, D_1 and D_2 are two constant depending on N and Q.

Proof: Suppose that [0,1] is partitioned into n_1 equispaced subintervals and S is the set of infimums of all intervals. We select n_2 level quantizer for the gray scale values and the set of quantization levels is called C. Finally for each interval I_x, $P(I_x)$ includes n_3 polynomial from $\mathcal{BP}_N^0([0,1],1)$. We generate a discrete set of functions D in the following way. We choose Q points from S in $\binom{n_1}{Q}$ different ways and call these points $\hat{d}_1, \hat{d}_2, \ldots \hat{d}_Q$. For each chosen point the grid point next to it will also be considered and these points are called $\tilde{d}_1, \tilde{d}_2, \ldots \tilde{d}_Q$. For any interval $[\tilde{d}_{i-1}, \hat{d}_i]$ we take 2 polynomials from $P([\tilde{d}_{i-1}, \hat{d}_i])$ and finally we assign a value from C to each region of each interval. For the other regions we can use the value 0. It can be verified that,

$$|D| = \sum_{i=1}^{|S|^{Q+1}} |C|^{3(Q+1)} |P([\tilde{d}_{i-1}, \hat{d}_i])|^{2(Q+1)} = {n_1}^{Q+1} {n_2}^{3Q+3} {n_3}^{2Q+2} \tag{9}$$

In order to keep the distortion less than ϵ we should have

$$\sup_{f \in \mathcal{T}_N^Q} \inf_{\hat{f} \in D} \|f - \hat{f}\|^2_{L^2([0,1]^2)} \leq \epsilon \tag{10}$$

note: explain how we are choosing $\hat{f}$ for a given f.

$$\|f-\hat{f}\|^2_{L^2([0,1]^2)} = \int_{x=0}^{1}\int_{y=0}^{1}(f-\hat{f})^2$$

$$\sum_i \int_{x=\tilde{d}_{i-1}}^{\hat{d}_i}\int_{y=0}^{1}(f-\hat{f})^2 + \sum_i \int_{x=\hat{d}_i}^{\tilde{d}_i}\int_{y=0}^{1}(f-\hat{f})^2 \leq$$

$$\sum_i \int_{x=\tilde{d}_{i-1}}^{\hat{d}_i}\int_{y=0}^{1}(f-\hat{f})^2 + \frac{Q}{n_1} \leq$$

$$\sum_i (\alpha_d-\hat{\alpha}_d)^2 \int_{x=\tilde{d}_{i-1}}^{\hat{d}_i} \min(h_d(x),\hat{h}_d(x)) +$$

$$\sum_i (\alpha_d-\hat{\alpha}_m)^2 \int_{x=\tilde{d}_{i-1}}^{\hat{d}_i} |h_d(x)-\hat{h}_d(x))| +$$

$$\sum_i (\alpha_m-\hat{\alpha}_d)^2 \int_{x=\tilde{d}_{i-1}}^{\hat{d}_i} |h_d(x)-\hat{h}_d(x))| +$$

$$\sum_i (\alpha_m-\hat{\alpha}_m)^2 \int_{x=\tilde{d}_{i-1}}^{\hat{d}_i} |\min(h_u(x),\hat{h}_u(x)) - \max(h_d(x),\hat{h}_d(x))| +$$

$$\sum_i (\alpha_u-\hat{\alpha}_u)^2 \int_{x=\tilde{d}_{i-1}}^{\hat{d}_i} |1-\max(h_u(x),\hat{h}_u(x))| +$$

$$\sum_i (\alpha_u-\hat{\alpha}_m)^2 \int_{x=\tilde{d}_{i-1}}^{\hat{d}_i} |h_u(x)-\hat{h}_u(x))| +$$

$$\sum_i (\alpha_m-\hat{\alpha}_u)^2 \int_{x=\tilde{d}_{i-1}}^{\hat{d}_i} |h_u(x)-\hat{h}_u(x))| + \frac{Q}{n_1} \leq$$

$$\frac{Q}{n_1} + 3(Q+1)(\frac{1}{n_2})^2 + 4(Q+1)K(\frac{1}{n_3})^{\frac{1}{N+1}} \tag{11}$$

In order to find an upper bound for the entropy, this optimization problem is considered.

$$\min {n_1}^{Q+1}{n_2}^{3Q+3}{n_3}^{2Q+2}$$
$$s.t. \frac{Q}{n_1} + 3(Q+1)(\frac{1}{n_2})^2 + 4(Q+1)K(\frac{1}{n_3})^{\frac{1}{N+1}} \leq \epsilon \tag{12}$$

the relaxed version of this discrete optimization problem is geometric programming problem [**?**]. By considering the following change of variables

$$z_i = \ln n_i,\ \ i=1,2,3, \tag{13}$$

the problem converts to this convex optimization problem

$$\min z_1 + 3z_2 + 2z_3$$
$$s.t.\ 2(Q+1)e^{-z_1} + \frac{3}{4}(Q+1)e^{-2z_2} + 4K(Q+1)e^{-\frac{z_3}{N+1}} \leq \epsilon$$

By using Karush-Kuhn-Tucker(KKT) condition we have

$$\begin{aligned}
&\lambda 2(Q+1)e^{-z_1} = 1\\
&\frac{1}{2}\lambda(Q+1)e^{-2z_2} = 1\\
&\lambda\frac{4K(Q+1)}{N+1}e^{\frac{-z_3}{N+1}} = 1.
\end{aligned}$$

where λ is the Lagrange multiplier. By combining these equations with 12

$$2(Q+1)e^{-z_1} + 3(Q+1)e^{-z_1} + 2(Q+1)(N+1)e^{-z_1} \leq \epsilon \tag{14}$$

and therefore,

$$\begin{aligned}
e^{-z_1} &\leq \frac{\epsilon}{(Q+1)(2N+7)}\\
e^{-z_2} &\leq \sqrt{\frac{4\epsilon}{(Q+1)(2N+7)}}\\
e^{-z_3} &\leq \left(\frac{\epsilon}{4K(Q+1)}\right)^{N+1}.
\end{aligned} \tag{15}$$

By using the optimal values for n_1, n_2 and n_3, the total number of balls can be found by,

$$\begin{aligned}
&{n_1}^{Q+1}{n_2}^{3Q+3}{n_3}^{2Q+2} =\\
&\left(\frac{(Q+1)(2N+7)}{\epsilon}\right)^{(Q+1)}\left(\frac{(Q+1)(2N+7)}{4\epsilon}\right)^{\frac{(Q+1)}{2}}\\
&\left(\frac{4K(Q+1)}{\epsilon}\right)^{2(Q+1)(N+1)} = C\left(\frac{1}{\epsilon}\right)^{\frac{(4N+7)(Q+1)}{2}}
\end{aligned} \tag{16}$$

and the upper bound of (8) is proved.

To find the lower bound of the ϵ-entropy, we consider a very special subclass of this space with the following properties.

$$\begin{aligned}
&\alpha_u = 1, \alpha_m = 0, \alpha_d = 1\\
&d_i = i/Q \ \forall i
\end{aligned} \tag{17}$$

we have,

$$\begin{aligned}
\|f - \hat{f}\|_{L^2(I_x\times I_y)} &= \|h_u(x) - \hat{h}_u(x)\|_{L^1(I_x)} + \|h_d(x) - \hat{h}_d(x)\|_{L^1(I_x)}\\
&\geq \|h_u(x) - \hat{h}_u(x)\|^2_{L^2(I_x)} + \|h_d(x) - \hat{h}_d(x)\|^2_{L^2(I_x)}
\end{aligned} \tag{18}$$

the last inequality comes from the fact that since $|h_u(x) - \hat{h}_u(x)| \leq 1$,

$$|h_u(x) - \hat{h}_u(x)| \geq |h_u(x) - \hat{h}_u(x)|^2 \tag{19}$$

Therefore the minimum number of balls that we need for coding this space, is greater than the total number of balls that we need for coding $h_u(x)$ if we keep the L_2-norm error greater less than ϵ and according to [**?**] paper this is greater than $C_1\left(\frac{1}{\epsilon}\right)^{(N+1)(Q+1)}$ ■

In the next section, we will discuss a practical framework to overcome complexity of the exhaustive search associated to the ϵ-entropy.

Theorem 6.2: The greedy partitioning algorithm is able to find the Q singularity points in at most $2Q+1$ steps. In other words, it will find $2Q+1$ partition points and Q of them are the actual singularity points.

Proof: Assume that at some stage of the algorithm, the partition point s_j is selected from interval $[d_i, d_{i+1})$ and this is the first time one of the partition points is in this interval. We want to prove that none of the other partition points will be in the interval (d_i, d_{i+1}). Assume that s_r is found to be the optimum selection point and $d_i < s_r < d_{i+1}$. Before selecting s_r the closest points to s_j were s_l and s_k. Without loss of generality assume that $s_l \leq s_r \leq s_j$. The total error of this approximation is:

$$\begin{aligned}\|g-\hat{g}\|^2_{L^2[s_l,s_j)\times I_y} &= \|g-f_1\|^2_{L^2[s_l,s_r)\times I_y} + \|g-f_2\|^2_{L^2[s_r,s_j)\times I_y} \\ &= \|g-f_1\|^2_{L^2[s_l,s_r)\times I_y} \end{aligned} \tag{20}$$

where $f_1 \in \mathcal{T}^0_N([s_l, s_r))$ and $f_2 \in \mathcal{T}^0_N([s_r, s_j))$. The second inequality comes from the fact that in the interval $[s_r, s_j)$, g can be approximated exactly and the distortion would be zero. The last term can also be written as:

$$\begin{aligned}\|g-f_1\|^2_{L^2([s_l,s_r)\times I_y)} &= \|g-f_1\|^2_{L^2([s_l,d_i)\times I_y)} + \|g-f_1\|^2_{L^2([d_i,s_r)\times I_y)} \\ &\geq \|g-f_1\|^2_{L^2([s_l,d_i)\times I_y)} \end{aligned} \tag{21}$$

and the equality is achievable if and only if $d_i = s_r$ or $f_1 = f_2$. But if $f_1 = f_2$ the algorithm will not divide the interval any more (because the cost function has the notion of Rate as well). Therefore, the only possible case is $d_i = s_r$ and between any two singularity points the algorithm will at most find one extra point. The lemma is proved.

■

Theorem 6.3: For a given $g \in \mathcal{T}^Q_N(I_x)$, the algorithm above which greedily selects elements from $\mathcal{D}^Q_N(I_x)$, finds its characterizing parameters (h_u, h_d, α_u, α_m, α_d) in only two steps.

Proof: First we note that

$$\mathbb{1}_{\{y\leq h_1(x)\}}\mathbb{1}_{\{y>h_1(x)\}} = 0$$

everywhere. Therefore, the mean-squared error between a function in $\mathcal{D}^Q_N(I_x)$ defined as

$$f_1(x,y) = \beta_{d1}\mathbb{1}_{\{y\leq h_1(x)\}} + \beta_{u1}\mathbb{1}_{\{y>h_1(x)\}}. \tag{22}$$

and the function $g \in \mathcal{T}^Q_N(I_x)$ is minimized when

$$\beta_{d1} = \frac{\displaystyle\int_I g\mathbb{1}_{\{y\leq h_1(x)\}}\,\mathrm{d}x\mathrm{d}y}{\|\mathbb{1}_{\{y\leq h_1(x)\}}\|^2_{L^2(I)}} = \frac{\displaystyle\int_{\{y\leq h_1(x)\}} g\,\mathrm{d}x\mathrm{d}y}{\displaystyle\int h_1(x)\mathrm{d}x}$$

$$\beta_{u1} = \frac{\displaystyle\int_I g\mathbb{1}_{\{y>h_1(x)\}},\mathrm{d}x\mathrm{d}y}{\|\mathbb{1}_{\{y>h_1(x)\}}\|^2_{L^2(I)}} = \frac{\displaystyle\int_{\{y>h_1(x)\}} g\,\mathrm{d}x\mathrm{d}y}{1-\displaystyle\int h_1(x)\mathrm{d}x}$$

where $|\cdot|$, with an abuse of notation, denotes the size of a set. The MSE can be written as

$$\begin{aligned}\|g-f_1\|^2_{L^2(I)} &= \|g-\beta_{d1}\mathbb{1}_{\{y\leq h_1(x)\}} - \beta_{u1}\mathbb{1}_{\{y>h_1(x)\}}\|^2_{L^2(I)} \\ &= \|g\|^2_{L^2(I)} + \beta^2_{d1}\int h_1(x)\mathrm{d}x + \beta^2_{u1}\left(1-\int h_1(x)\mathrm{d}x\right) \\ &- 2\beta_{d1}\int_I g\mathbb{1}_{\{y\leq h_1(x)\}} - 2\beta_{u1}\int_I g\mathbb{1}_{\{y>h_1(x)\}} \\ &= \|g\|^2_{L^2(I)} - \Theta \end{aligned}$$

where

$$\Theta = \beta_{d1}^2 \int h_1(x)\mathrm{d}x - \beta_{u1}^2 \left(1 - \int h_1(x)\mathrm{d}x\right) \tag{23}$$

By substituting g as

$$g(x,y) = \alpha_d \mathbb{1}_{\{y \le h_d(x)\}} + \alpha_m \mathbb{1}_{\{h_d(x) < y \le h_u(x)\}} + \alpha_u \mathbb{1}_{\{y > h_u(x)\}}, \tag{24}$$

we rewrite (25) as

$$\begin{aligned} \Theta &= \frac{\left(\int_{\{y \le h_1(x)\}} g \,\mathrm{d}x\mathrm{d}y\right)^2}{\int h_1(x)\mathrm{d}x} + \frac{\left(\int_{\{y > h_1(x)\}} g \,\mathrm{d}x\mathrm{d}y\right)^2}{1 - \int h_1(x)\mathrm{d}x} \\ &= \frac{\left(\alpha_d r_{dd} + \alpha_m r_{md} + \alpha_u r_{ud}\right)^2}{r_{dd} + r_{md} + r_{ud}} + \frac{\left(\alpha_d r_{du} + \alpha_m r_{mu} + \alpha_u r_{uu}\right)^2}{r_{du} + r_{mu} + r_{uu}} \end{aligned} \tag{25}$$

where (see Figure 4)

$$\begin{aligned} r_{dd} &= \int \min[h_1(x), h_d(x)]\mathrm{d}x, \\ r_{md} &= \int \min[h_1(x), h_u(x)] - \min[h_1(x), h_d(x)]\mathrm{d}x, \\ r_{ud} &= \int h_1(x) - \min[h_1(x), h_u(x)]\mathrm{d}x, \\ r_{du} &= \int 1 - \max[h_1(x), h_u(x)]\mathrm{d}x, \\ r_{mu} &= \int \max[h_1(x), h_u(x)] - \max[h_1(x), h_d(x)]\mathrm{d}x, \\ r_{uu} &= \int \max[h_1(x), h_d(x)] - h_1(x)\mathrm{d}x. \end{aligned}$$

We note that

$$\begin{aligned} r &= r_{dd} + r_{md} + r_{ud} + r_{du} + r_{mu} + r_{uu} = \int_I \mathrm{d}x\mathrm{d}y \\ \alpha &= \alpha_d(r_{dd} + r_{du}) + \alpha_m(r_{md} + r_{mu}) + \alpha_d(r_{ud} + r_{uu}) \\ &= \int_I g(x,y)\mathrm{d}x\mathrm{d}y \end{aligned}$$

are constants. After some algebra, (23) will be

$$\Theta = \frac{\left((\alpha_d - \alpha/r)r_{dd} + (\alpha_m - \alpha/r)r_{md} + (\alpha_u - \alpha/r)r_{ud}\right)^2}{(r_{dd} + r_{md} + r_{ud})(r - r_{dd} - r_{md} - r_{ud})} - \alpha^2 \tag{26}$$

To minimize MSE (or maximize Θ), we should take the following constraints into account

$$\begin{aligned} 0 \le \quad r_{dd} \quad &\le r_d = \int h_d(x)\mathrm{d}x \\ 0 \le \quad r_{md} \quad &\le r_m = \int h_u(x) - h_d(x)\mathrm{d}x \\ 0 \le \quad r_{ud} \quad &\le r_u = \int 1 - h_u(x)\mathrm{d}x \end{aligned}$$

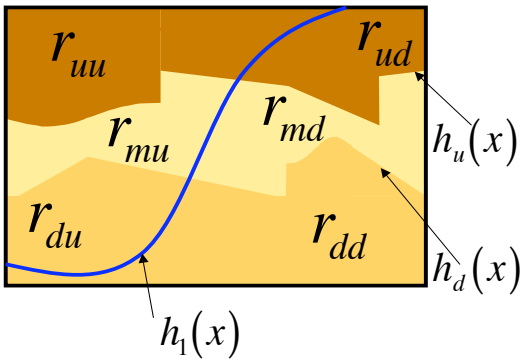

Fig. 4. Intersections of a function from $\mathcal{D}_N^Q(I_x)$ with a function from $\mathcal{T}_N^Q(I_x)$ and resulted regions

which in fact forms a cuboid (hyperrectangle) in 3-D space of (r_{dd}, r_{md}, r_{ud}). It can be easily shown that partial derivatives of Θ are non-zero within this cuboid. Proof is trivial for the case where $\alpha_d = \alpha_m = \alpha_u$ or where any of the values in triplet (r_d, r_m, r_u) is zero. For the non-trivial case, the maximum of Θ is only attainable at a vertex except at the vertex (r_d, r_m, r_u) (since at least one of the coefficients in the numerator (26) will be strictly negative). This implies that f_1 minimizes the MSE if its characterizing function $h_1(x)$ coincides with either h_d or h_u. Clearly, once one of these functions is identified, the other one will be captured at the second step. ■