ϵ-Entropy of Piecewise Polynomial Functions and Tree Partitioning Compression

Arian Maleki
Department of Electrical Engineering
Stanford University
arianm@stanford.edu

Gunnar Carlsson
Department of Mathematics
Stanford University
gunnar@math.stanford.edu

***Abstract*—Most of the signals in nature are piecewise smooth. One of the simple and yet efficient models for representing smooth signals is the class of piecewise polynomials. In this paper compression of this class of functions is considered. Some bounds are derived for the ϵ-entropy of this class of functions. These bounds show us the best performance the optimum compression scheme can have. By comparing it with the performance of traditional binary trees, it is demonstrated that the rate-distortion behavior of binary tree is far from optimum. We will then show that a simple modification of binary trees results in much better performance binary tree algorithms. This modification will retain all the advantages of binary trees.**

Index Terms- Entropy, signal, data compression, quadtree, piecewise polynomial

I. Introduction

A. Problem and Framework

Most of the signals in nature are piecewise smooth. One of the simple and yet efficient models for representing smooth signals is the class of piecewise polynomials [3], [4], [6]. For example this class of functions can be used to model the edges in natural and cartoon images [1], [13] . Therefore, a good representation and a good compression scheme for this class of functions will lead us to good models for different types of signals. In this paper, this class of functions is studied. The main goal is to find a simple and practical compression scheme with close to optimum performance. In order to analyze each compression scheme we need the following: 1) A framework for measuring the quality of each algorithm, and 2) the performance of the optimum algorithm in this framework. There are at least two frameworks in the approximation theory and information theory literature. One of them is called approximation power [14] and the second one is called rate-distortion (or equivalently distortion-rate) behavior [2], [9]. Let F denote a class of functions. D_F is called a dictionary for F if and only if any element of F can be written as a linear combination of the elements of D_F. In nonlinear approximation, the asymptotic behavior of deviation from the best k term approximation is a measure of quality for each dictionary D_F. Although approximation power may work for comparison of two orthonormal bases (like wavelet and Fourier for L^2 functions), it is not the right framework for overcomplete dictionaries, since the cardinalities of the dictionaries do not play any role. Rate-distortion theory addresses this problem by considering rate and distortion simultaneously, where rate is the total number of bits used to represent the compressed signal and distortion is the distance between the original and compressed signal. There are two closely related rate-distortion theories: Shannon's rate-distortion [9] and Kolmogorov's ϵ-entropy [7]. Shannon's theory deals with stochastic sources. It ignores the processes that have very small probabilities and just codes the rest. Kolmogorov's ϵ-entorpy, on the other hand, deals with deterministic sources and enforce the algorithm to code the signals such that the distortion of any signal after compression is less than the desired level. Since we do not have any probabilistic model for the space of piecewise polynomials and all of the functions are equally important in our analysis, we will use Kolmogorov's ϵ-entropy instead of Shannon's well known rate-distortion theory. It should also be mentioned that there is an interesting connection between these two theories which is explained in [10].

B. Related Work

There exists some related work in the literature as well. Prandoni et al. [3] derived the rate distortion behavior of an oracle method and proved that traditional bases such as wavelet and Fourier are not optimum for piecewise polynomial functions. They have also proposed a dynamic programming algorithm to implement their oracle method. But this algorithm suffers from a few problems. First, although Prandoni's algorithm uses dynamic programming, the computational complexity is still very high. Second, it cannot be extended to higher dimensional signals. Third, it uses some prior knowledge about the number of polynomial pieces which is usually not available. In order to address these problems binary tree partitioning (or in two dimensions quad-tree partitioning [1]) algorithm with lagrangian cost function has been proposed [4]. This algorithm has some advantages: 1) it is simple and efficient. 2) Some automatic methods have been proposed to set the value of parameters for a given achievable rate-distortion [15]. 3) By choosing the dictionaries properly, they will result in higher dimensional coding schemes [1], [13]. 4) They are very flexible in the sense that they can be easily modified to lead to new compression schemes. In spite of all these advantages, it will be shown that the performance of binary partitioning(derived in [4]) is far from the upper bound that we will derive for ϵ-entropy. Shukla et al. [4] improved the performance of binary trees by adding a join step to the algorithm. But, in order to have all the advantages of binary partitioning it is more interesting to modify binary partitioning algorithm, instead of adding a second stage to it. Furthermore, by using the join step we will loose the automatic parameter selection. Our modification of binary trees consists of permitting the division of intervals into two subintervals of nonequal length. It should be mentioned that the join step can also be used for the new binary trees as well.

The organization of this paper is as follows: In section II, the concept of Kolmogorov's ϵ-entropy for compact spaces is reviewed. Then some bounds for the ϵ-entropy of piecewise polynomial functions will be derived. In section III, the binary partitioning algorithm will be discussed in a general framework. Also the limitations of the traditional binary trees used in signal processing will be discussed in this section. In section IV a greedy partitioning algorithms will be introduced. In spite of its greedy nature it has some optimality properties that will be discussed in that section.

II. Kolmogorov's ϵ-Entropy of the class of piecewise polynomial functions

Let $F \subset L^2([0,1])$ be a compact class of functions. The goal is to compress the functions in this space such that the distortion (the L^2-norm of the difference between the original and the compressed function) is less than ϵ and the question is: what is the least number of bits needed to code the functions in this space? The answer to this question is given by Kolmogorov [7]. The *ϵ-entropy* of a compact set F is shown with $H_\epsilon(F)$ and is defined as

$$H_\epsilon(F) = \log_2 N_\epsilon(F), \tag{1}$$

where $N_\epsilon(F)$ is the minimal number of sets in an ϵ-covering [8], [7], [12] of F.

As may be noted, for coding an element of F with the distortion at most equal to ϵ the best achievable rate is $H_\epsilon(F)$. Consider the space of piecewise polynomial functions on an interval I to be represented by $Poly_N^Q(I)$ in which N is the maximum degree of the polynomials on this interval and Q is the maximum number of singularities. A singularity is a point at which the function is not infinitely differentiable. Let $BPoly_N^Q(I,A)$ denote the set of all piecewise polynomials of maximum degree N over I with Q singularities which are bounded by A, i.e. ,

$$BPoly_N^Q(I,A) = \{f \in Poly_N^Q(I) \mid \sup_{x\in I}|f| \leq A\}. \tag{2}$$

We have proved in [5] that $BPoly_N^Q(I,A)$ is compact. Therefore ϵ-entropy can be defined for this space. In the following theorems L_2 norm is considered as the metric.

Theorem 2.1: There exist two positive constants B_1 and B_2 such that the epsilon entropy of the space $BPoly_N^0([0,1],A)$ satisfies the following inequalities

$$B_1 + \log(\frac{1}{\epsilon}) \leq \frac{1}{N+1} H_\epsilon(BPoly_N^0([0,1],A)) \leq B_2 + \log(\frac{1}{\epsilon}),$$

where $B_2 = \log\left(\frac{A\sqrt{N+1}}{2}\right)$ and $B_1 = \log\left(\frac{A}{C}\right)$, in which C is the constant appearing in [5].

Proof: The proof of this theorem is given in [5]. It should be mentioned that since we use these bounds in the high bit rate (or low distortion) regime, the constants B_1 and B_2 are not important. ■

Theorem 2.2: The ϵ-entropy of the space $BPoly_N^Q([0,1],A)$ satisfies the following two constraints

$$C_1 + (N+1)(Q+1)\log(\frac{1}{\epsilon}) \leq H_\epsilon(BPoly_N^Q([0,1],A)) \leq C_2 + (N+3)(Q+1)\log\lceil\frac{1}{\epsilon}\rceil \tag{3}$$

where,

$$C_1 = (N+1)(Q+1)\log\left(\frac{A}{C'}\right),$$
$$C_2 = \lceil (N+3)(Q+1)\log\left(\frac{A(Q+1)(N+3)}{(2\sqrt{N+1})^{(N+1)/(N+3)}}\right)\rceil$$

Proof: In order to prove the upper bound, assume that the $[0,1]$ is partitioned into n_1 equispaced subintervals. We choose Q points out of these n_1 points in $\binom{n_1}{Q}$ different ways. For each chosen point, the point next to it will also be considered. Let's call these points $0 = \hat{d}_0, \hat{d}_1, \ldots, \hat{d}_{2Q+1} = 1$. For each interval $[\hat{d}_{2i}, \hat{d}_{2i+1})$, n_2 polynomials will be picked from all the polynomials between the selected points. The way these polynomials are chosen is explained in the proof of theorem 2.1 in [5]. As a brief explanation, these points are placed on a uniform grid in this N-dimensional subset. Therefore, the total number of functions(balls) we consider is: $\binom{n_1}{Q}\ {n_2}^{Q+1}$. Let D be the set of all these functions and f be an arbitrary function in the space $BPoly_N^Q([0,1],A)$. The distortion in approximating this function is $\min_{\hat{f}}\|f-\hat{f}\|$ where $\hat{f} \in D$. The distortion can be upper bounded by:

$$\|f(x)-\hat{f}(x)\|_{L^2(I)} \leq \sum \|f(x)-\hat{f}(x)\|_{L^2[\hat{d}_{2i+1},\hat{d}_{2i+2}]} + \sum \|f(x)-\hat{f}(x)\|_{L^2[\hat{d}_{2i},\hat{d}_{2i+1}]} \tag{4}$$

in which $\|f\|_{L^2(I)} = (\int_I f^2)^{\frac{1}{2}}$.

Assume that $\hat{f}$ is chosen from D with the following method: First the decision is made on the $\hat{d}_{2i-1}$ points of $\hat{f}$. $\hat{d}_{2i-1}$ is chosen such that it is less than or equal to d_i and among all the points on the $1/n_1$ grid that satisfy this condition, is the closest to the point d_i. Therefore we have: $|d_i - \hat{d}_{2i-1}| \leq 1/n_1$ and $d_i \leq \hat{d}_{2i}$ since $\hat{d}_{2i} = \hat{d}_{2i-1} + 1/n_1$. Then in the intervals $[\hat{d}_{2i}, \hat{d}_{2i+1}]$ find the closest polynomial in that part that is also part of D. With this choice of approximation, each term on the second line of (4) is less than or equal to $2A\sqrt{1/n_1}$. Also according to the previous theorem by choosing the polynomials properly, the error of the terms on the third line could be made less than $K(\frac{1}{n_2})^{\frac{1}{N+1}}$, where $K = \frac{A}{2}\sqrt{N+1}$. Therefore, the total error is less than or equal to:

$$\|f-\hat{f}\| \leq (Q+1)K(\frac{1}{n_2})^{\frac{1}{N+1}} + 2(Q+1)A(\frac{1}{n_1})^{\frac{1}{2}} \leq \epsilon \tag{5}$$

The last inequality is imposed in order to keep the distortion less than ϵ. The number of balls is equal to $\binom{n_1}{Q}\ {n_2}^{Q+1}$. The goal is to find an upper bound for the number of balls and the number of balls can be upper bounded by ${n_1}^{Q+1}{n_2}^{Q+1}$. With this approximation the problem is simplified to:

$$\begin{aligned} \text{minimize} \quad & n_1^{Q+1} n_2^{Q+1} \\ \text{s.t.} \quad & (Q+1)\left[K(\frac{1}{n_2})^{\frac{1}{N+1}} + \frac{2A}{\sqrt{n_1}}\right] \leq \epsilon \end{aligned} \tag{6}$$

Instead of solving this problem for integer numbers, we assume that n_1 and n_2 are not necessarily integers. Then the results will be rounded to the closest integer above them. The new problem is geometric programming [11]. By changing the variables and using KKT conditions upper bound can be found.

For proving the lower bound consider a subset of $BPoly_N^Q([0,1],A)$ consisting of all functions in $BPoly_N^Q([0,1],A)$ that have singularities at $\{\frac{1}{Q+1}, \frac{2}{Q+1}, \ldots \frac{Q}{Q+1}\}$. Since this is a $(N+1)(Q+1)$ dimensional space, according to the equivalence of L_∞ and L_2 norms of any finite dimensional space, i.e. ,

$$c'\|f\|_{L2} \leq \|f\|_{L_\infty} \leq C'\|f\|_{L2}, \tag{7}$$

and the fact that the L_∞-norm of these signals is bounded, the L_2 ball $\{f \mid \|f\|_{L_2} \leq \frac{A}{C'}\}$ is included in this space. Using the same argument used in theorem 2.1, at least $2^{C_1}(\frac{1}{\epsilon})^{(N+1)(Q+1)}$ will be necessary to code this L_2 ball, where

$$C_1 = (N+1)(Q+1)\log\left(\frac{A}{C'}\right), \tag{8}$$

and the proof of the theorem is complete. ■

In the last theorem, ϵ is playing the role of maximum distortion and the number of balls is related to the number of bits that should be used for compressing the signal at the ϵ distortion level. But it is also possible to look at this problem from a different point of view. Assume that the number of bits that can be used for representing the signals is fixed at some bit rate R and the goal is to minimize the distortion.

Theorem 2.3: For a fixed number of bits R the least achievable distortion satisfies the following inequalities

$$k_1 2^{\frac{-R}{(N+1)(Q+1)}} \leq D(R) \leq k_2 2^{\frac{-R}{(N+3)(Q+1)}}, \tag{9}$$

where $k_2 = A(Q+1)(N+3)$ and $k_1 = \frac{A}{C'}$.

Proof: It follows directly from theorem 2.2, by noting that $\left(2\sqrt{N+1}\right)^{(N+1)/(N+3)} \geq 1$. ■

Since the upper bound is found by an exhaustive search through a very large dictionary, it is not practical. The goal of the next two sections is to find a good method for compressing these kinds of signals. The desired method should have the good characteristics mentioned in the introduction.

III. Binary Partitioning Algorithm

As explained in the introduction, wavelet and the other traditional algorithms cannot perform well on this class of functions. In order to solve the problem Shukla et al. [4] have proposed an interesting algorithm which is called Binary Tree Partitioning. The idea of this algorithm is somewhat similar to the idea of wedgelet [1]. By using binary partitioning or quad partitioning these two algorithms partition the signal into some pieces and then code each piece with a polynomial (or wedge like patches in wedgelet). The complete description of the algorithm is as follows: Let $P^j = \{I_1^j, I_2^j, \ldots, I_L^j\}$ be a partition of the interval $[0,T)$ at level j with the following characteristics: First, each $I_i^j \subset [0,T)$ is a half open interval e.g., $I_i^j = [d_{i-1}, d_i)$. Second,

$$\bigcup_k I_k^j = [0,T), \quad I_k^j \bigcap I_i^j = \oslash \quad \forall k \neq i \tag{10}$$

For each interval I_k^j a cost function is defined as:

$$c_k^j = \left\| [x(t) - P_o(t)] 1_{I_k^j}(t) \right\|_{L_2([0,T))} + \lambda^2 R_k^j \tag{11}$$

where $P_o(t)$ is the best polynomial of degree N that minimizes the first term of the cost function, R_k^j is the required rate for coding that part of the tree and λ is a real number. The binary tree at level j, divides each I_k^j with $|I_k^j| = T2^{-j}$ at its midpoint to two smaller intervals I_{2k-1}^{j+1} and I_{2k}^{j+1} if:

$$c_k^j > c_{2k-1}^{j+1} + c_{2k}^{j+1} \tag{12}$$

By starting from $P^0 = \{[0,T)\}$ and applying the same algorithm to the partitions iteratively one will get to the point that none of the partitions can be divided more. The final partition is the one which is used in coding. The intervals that are divided into two pieces are called internal nodes and the intervals that cannot be divided are called terminal nodes or leaves. This terminology will be used in this paper. Let me summarize the binary tree partitioning algorithm in a different way that will lead us to a more general framework. Assume that at each scale j, a set of signals is given $\Phi^j = \{\phi_k^j\}_{k=1}^{K_j}$. At each level the best thing one can do is to replace a signal in one of the intervals with one of the signals in the set corresponding to the scale of that signal. In the case of binary partitioning algorithm explained before each set includes a discrete version of the polynomials of degree N that are defined in the interval of size $T2^{-j}$. It can be proved that for this algorithm $D_{BT}(R) \geq \alpha_0 \sqrt{R} 2^{-\alpha_1 \sqrt{R}}$ which is far from the ϵ-entropy bounds(the rate of decay here is $2^{-\alpha\sqrt{R}}$ and $2^{-\beta R}$ in the entropy bounds). Since the proof of this theorem is very similar to the proof given in [4] we ignore this proof. In [5] it is proved that by enlarging Φ^j one can get better rate distortion characteristic from the binary tree algorithm. In this paper we will consider a different modification of the binary tree algorithm. The question that will be answered in the next section is that what happens in the rate-distortion sense if we generalize the idea of binary partitioning?

IV. Generalized Binary Partitioning

In this section the goal is to generalize the binary partitioning algorithm explained in the last section in order to improve its performance. We will show that although the algorithm that is proposed here is greedy in its nature, its rate-distortion characteristic is very close to the upper bound of ϵ-entropy. Because of its greedy nature it is very fast and because of its tree structure it can be easily extended to higher dimensional signals (the partitions will be rectangular). In the binary tree partitioning algorithm which was explained in the previous section each partition at each level is divided into two intervals with the same size (this is the case if the cost function is improved). But one can divide the interval into two nonequal subintervals. This kind of binary partitioning is called generalized binary partitioning. In coding a signal by this method, the partition points should also be coded. In order to explain the procedure let's assume that at some level j an interval $I_k^j = [t_{k_{start}}^j,\, t_{k_{stop}}^j)$ exists. For a given partition point $p \in I_k^j$, I_k^j can be partitioned into these two subintervals: $I_{k_{p<}}^{j+1} = [t_{k_{start}}^j,\, p)$ and $I_{k_{p>}}^{j+1} = [p,\, t_{k_{stop}}^j)$;

The problem of finding the partition point p is:

$$(p^*, P_1^*, P_2^*) = \arg \min_{p \in I_k^j, P_1 \in \Phi(I_{k_{p<}}^{j+1}), P_2 \in \Phi(I_{k_{p>}}^{j+1})} \|f(t) - P_1(t) 1_{I_{k_{p<}}^{j+1}} - P_2(t) 1_{I_{k_{p>}}^{j+1}}\| \tag{13}$$

where Φ is the set of signals. The next lemma shows that although the algorithm is greedy it will find the Q singularity points in at most $2Q+1$ steps.

Lemma 4.1: Let the original function f be in $BPoly_N^Q(I, A)$. Suppose that we have infinite precision in choosing the singularity points. If $|\lambda|$ is zero and Φ is the set of polynomials of degree N, the maximum number of partition points is $2Q+1$.

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:

$$\begin{aligned} \|f - \hat{f}\|_{L^2[s_l, s_j)}^2 &= \|f - P_1\|_{L^2[s_l, s_r)}^2 + \|f - P_2\|_{L^2[s_r, s_j)}^2 \\ &= \|f - P_1\|_{L^2[s_l, s_r)}^2 \end{aligned} \tag{14}$$

In which P_1 and P_2 are two polynomials. The second inequality comes from the fact that in the interval $[s_r, s_j)$, f is a polynomial and the distortion would be zero. The last term can also be written as:

$$\begin{aligned} \|f - P_1\|_{L^2[s_l, s_r)}^2 &= \|f - P_1\|_{L^2[s_l, d_i)}^2 + \|f - P_1\|_{L^2[d_i, s_r)}^2 \\ &\geq \|f - P_1\|_{L^2[s_l, d_i)}^2 \end{aligned} \tag{15}$$

and the equality is achievable if and only if $d_i = s_r$ or $P_1 = P_2$. But if $P_1 = P_2$ 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. ■

Lemma 4.2: Let the original function f be in $BPoly_N^Q(I, A)$. Suppose that we have finite precision in choosing the singularity points. If $|\lambda|$ is zero and Φ is the set of polynomials of degree N, the maximum number of partition points will be $3Q + 1$.
The proof of this lemma is very similar to the proof of the previous one. The only thing that should be mentioned here is that $2Q$ points of these $3Q+1$ points are the points closest to the singularity points of the main function and therefore $2Q+1$ of these points are independent and should be coded for the encoder.
The last thing that should be mentioned here, is that since the encoder solves a least squares problem to find the polynomials in each interval, it does not need to do exhaustive search and also the encoder will have the coefficients of the polynomials exactly (although it quantizes the coefficients for transmission).

Theorem 4.3: If the original function f is in $BPoly_N^Q([0, T], A)$ the distortion rate behavior of generalized binary partitioning with polynomial dictionary is upper bounded by:

$$D(R) \leq c2^{-\frac{R}{2(N+3)(Q+1)}} \tag{16}$$

Proof: The complete coding scheme includes coding the structure of the partitions (or partition points) in addition to the polynomials in each partition. Let $s_0, s_1, s_2, \ldots s_{2Q+1}, s_{2Q+2}$ be the partition points, where $s_0 = 0$ and $s_{2Q+2} = T$. In order to code these selection points we divide the interval $[0, T]$ into 2^{R_s} equispaced partitions. When a selection point is in one of these intervals, it is approximated by the infimum of that interval. Let $\hat{s}_1, \hat{s}_2, ..., \hat{s}_{2Q+1}$ represent the coded partition points. The total error is:

$$\|f - \hat{f}\|_{L^2[0,T]} \leq \sum_{i=0}^{2Q+1} \|f - \hat{f}\|_{L^2[s_i,\hat{s}_{i+1})} + \sum_{i=0}^{2Q+1} \|f - \hat{f}\|_{L^2[\hat{s}_i,s_i)} \tag{17}$$

If we code each polynomial by R_{p_i} bits, then each term in the first summation is bounded by [5],

$$\|f - \hat{f}\|_{L^2[s_i,\hat{s}_{i+1})} \leq \frac{A}{2}|I_i|^{\frac{1}{2}}(N + 1)^{\frac{1}{2}} 2^{\frac{-R_{p_i}}{N+1}} \tag{18}$$

The terms in the second summation of (17) can be divided into two groups. Q of them correspond to the singularity points and $Q + 1$ of them are some points in between. If one of the singularity points (which is the i'th selection point) is coded with R_s bits the resulting distortion would be bounded with:

$$\|f - \hat{f}\|_{L^2[\hat{s}_i,s_i)} \leq 2AT^{\frac{1}{2}} 2^{-\frac{R_s}{2}} \tag{19}$$

But the error in the other selection points doesn't change the total error. Therefore the total distortion of compression is:

$$\|f - \hat{f}\|_{L^2[0,T]} \leq \sum_{i=0}^{2Q+1} \frac{A}{2} T^{\frac{1}{2}}(N + 1)^{\frac{1}{2}} 2^{\frac{-R_{p_i}}{N+1}} + Q2AT^{\frac{1}{2}} 2^{-\frac{R_s}{2}} \tag{20}$$

In order to get better rate distortion characteristic this expression should be minimized subject to the bit rate constraint:

$$2QR_s + \sum_{i=1}^{2Q+1} R_{p_i} \leq R \tag{21}$$

KKT conditions will result in $R_{p_i} = R_{p_j} = R_p \ \forall i, j$, and:

$$(\frac{R_p}{N + 1}) - \frac{R_s}{2} = \alpha \tag{22}$$

in which $\alpha = \frac{1}{2} \log_2 \frac{1}{N+1}$; This equation is true for high bit rate regime and for low bit rate regime R_s should be equal to zero (water filling). By some simple calculations we will reach (16).

■

Although the performance of this algorithm is not the same as the upper bound of rate-distortion, it is close. There are some other ways to improve the performance of the algorithm. For example by joining the partitions [4] of this tree, it can be shown that the performance will be the same as the upper bound of ϵ-entropy.

V. Conclusion

In this paper compression of the class of piecewise polynomials is considered. The importance of this class is due to the fact that most of the signals in nature are piecewise smooth and piecewise smooth signals can be approximated with the desired precision with one of the signals in this class. Some bounds for the Kolmogorov's ϵ-entropy are calculated. These bounds show the limits of compression. Since the performances of the traditional compression algorithms such as wavelet and Fourier are far from the optimal distortion rate behavior, we proposed a new method by modifying the binary tree partitioning algorithm in order to get closer to the bounds. The proposed algorithm is the generalized binary tree partitioning. It is shown in the paper that in spite of its greedy nature, this algorithm has some optimality properties and can detect the singularity points correctly in a short number of trials (two times the number of discontinuities).

Acknowledgement

The first author would like to acknowledge DARPA grant, HR0011-05-1-0007.

References

[1] D. Donoho, "Wedgelets: nearly minimax estimation of edges,"Annals of Stat., vol. 27, pp. 859-897, 1999.
[2] M. Vetterli, "Wavelets, approximation and compression", IEEE Signal Processing Magazine, 2001.
[3] P. Prandoni, M. Vetterli, "Approximation and compression of piecewise smooth functions," Phil. Trans. Roy. Soc. London, vol. 357, no. 1760, pp. 2573-2591, Sep. 1999.
[4] 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.
[5] A. Maleki, "Piecewise polyomial functions and binary partitioning," Asilomar Conference on Signals, Systems and Computers, 2007, accepted for publication.
[6] M. N. Do, P. L. Dragotti, R. Shukla, M. Vetterli, "On the compression two-dimensional piecewise smooth functions ," Proceedings of IEEE International Conference on Image Processing, Thessaloniki, Greece, Oct. 2001, pp. 14-17.
[7] 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.
[8] H. L. Royden, Real Analysis,Second Edition, New york:Macmillan, 1968.
[9] T. M. Cover, J. A. Thomas, Elements of Information Theory, New York, Wiley, 1991.
[10] D. L. Donoho, "Wald lecture 1: counting bits with Shannon and Kolmogorov", Technical Report, Stanford University, 2000.
[11] S. Boyd, L. Vandenberghe, Convex Optimization, Cambridge University Press, First edition, 2004.
[12] A. G. Vitushkin, Theory of The Transmition and Processing of Information, Pergamon Press, 1961.
[13] V. Chandrasekaran, M. B. Wakin, D. Baron, R.G. Baraniuk, "Representation and compression of multi-dimensional piecewise functions using surflets", submitted to IEEE Transaction on Information Theory, 2006.
[14] R. Devore, Nonlinear Approximation, Acta Numerika (1998),51-150.
[15] K. Ramchandran, M. Vetterli, "Best wavelet packet bases in a rate-distortion sense", IEEE Transaction on Image Processing, vol. 2,pp. 160-175, 1993.