Approximation and Compression of Piecewise Smooth Functions

By Paolo Prandoni[1] and Martin Vetterli[1,2]
[1] Laboratory for Audio-Visual Communications
Swiss Federal Institute of Technology CH-1015 Lausanne, Switzerland
[2] Department of Electrical Engineering and Computer Sciences,
University of California, Berkeley, CA 94720

Wavelet or subband coding has been quite successful in compression applications, and this success can be attributed in part to the good approximation properties of wavelets. In this paper, we revisit rate-distortion bounds for wavelet approximation of piecewise smooth functions, in particular the piecewise polynomial case. We contrast these results with rate-distortion bounds achievable using an oracle based method. We then present a practical dynamic programming algorithm achieving performance similar to the oracle method, and present experimental results.

1. Introduction

Wavelets have had an important impact on signal processing theory and practice. In particular, wavelets play a key role in compression, image compression being a prime example. This success is linked to the ability of wavelets to capture efficiently both stationary and transient behaviors. In signal processing parlance, wavelets avoid the problem of the window size (as in the short-time Fourier transform, for example), since they work with many windows due to the scaling property.

An important class of processes encountered in signal processing practice can be thought of as "piecewise stationary". As an example, speech is often analyzed using such a model, for example in local linear predictive modeling as used in speech compression. Such processes can be generated by switching between various stationary processes. Wavelet methods are possible models as well, being able to fit both the stationary part and capture the breakpoints.

In the deterministic case, "piecewise smooth functions" are a class of particular interest. For an example, consider piecewise polynomial functions. Again, wavelet are good approximants if simple non-linear approximation schemes are used. The performance of wavelets in such a context is again linked to their ability to fit polynomials by scaling functions (up to the appropriate approximation order) while capturing the break points efficiently by a small number of wavelet coefficients.

When one is interested in compression applications, a key question is not just the approximation behavior, but the effective rate-distortion characteristic of schemes where wavelets and scaling functions are used as elementary approximation atoms.

The purpose of this article is first to review some recent work in wavelets and subband coding, recalling some now classic results on approximating piecewise smooth functions or piecewise stationary processes. Then, piecewise polynomial functions are specifically analyzed, and the differences between a wavelet based and an oracle based method are shown. Next, we present a practical method based on dynamic programming that performs just like the oracle method, and include some experimental results.

2. The Compression Problem

Compression is the trade-off between description complexity and approximation quality. Given an object of interest, or a class of objects, one studies this trade-off by choosing a representation (e.g. an orthonormal basis) and then deciding how to describe the object parsimoniously in the representation. Such a parsimonious representation typically involves approximation.

For example, for a function described with respect to an orthonormal basis, only a subset of basis vectors might be used (subspace approximation) and the coefficients used in the expansion are approximated (quantization of the coefficients). Thus, both the subspace approximation and the coefficient quantization contribute to the approximation error. More formally, for a function f in $L_2(R)$ for which $\{\varphi_n\}$ is an orthonormal basis, we have the approximate representation.

$$\hat{f} = \sum_{n \in I} \hat{\alpha} \varphi_n \tag{2.1}$$

$$\hat{\alpha}_n = Q[< \varphi_n, f >] \tag{2.2}$$

where I is an index subset and $Q[.]$ is a quantization function, like for example the rounding to the nearest multiple of a quantization step $\triangle$:

$$\hat{\alpha} = Q[\alpha] = \triangle \cdot \left(\left\lfloor \frac{\alpha}{\triangle} \right\rfloor + \frac{1}{2} \right). \tag{2.3}$$

Typically, the approximation error is measured by the L_2 norm, or squared distortion

$$\epsilon = \|f - \hat{f}\|_2^2 \tag{2.4}$$

The description complexity corresponds to describing the index set I, as well as describing the quantized coefficients $\hat{\alpha}_n$. The description complexity is usually called the rate R, corresponding to the number of binary digits (or bits) used. Therefore the approximation $\hat{f}$ of f leads to a rate-distortion pair (R, ϵ), indicating one possible trade-off between description complexity and approximation error. The example just given, despite its simplicity, is quite powerful and actually used in practical compression standards. It also raises the following questions:

Q1: What are classes of objects of interest and for which the rate-distortion trade-off can be well understood ?
Q2: If approximations are done in bases, what are good bases to use ?
Q3: How to choose the index set and the quantization ?

Q4: Are there objects for which approximation in bases is suboptimal ?

Historically, Q1 has been addressed by the information theory community in the context of rate-distortion theory. Shannon posed the problem in his 1948 landmark paper and proved rate-distortion results in his 1959 paper. The classic book by Berger (1971) is still a reference on the topic. Yet, rate-distortion theory has been mostly concerned with exact results within an asymptotic framework (the so-called large blocksize assumption together with random coding arguments). Thus, only particular processes (e.g. jointly Gaussian processes) are amenable to this exact analysis. But the framework has been used extensively, in particular in its operational version (when practical schemes are involved), see for example the review by Ortega and Ramchandran (1998). It is to be noted that rate-distortion analysis covers all cases (e.g. small rates with large distortions) and that the case of very fine approximation (or very large rates) is usually easier but less useful in practice.

The second question has a simple answer, based on rate-distortion theory, in the stationary jointly Gaussian case. Then, the canonical basis is the Karhunen-Loève basis, and a procedure called reverse waterfilling leads to the optimal behavior. Yet, not all things in life are jointly Gaussian, and that is where wavelets come into play. For processes which are piecewise smooth (e.g. images), the abrupt changes are well captured by wavelets, and the smooth or stationary parts are efficiently represented by coarse approximations using scaling functions. Both practical algorithms (e.g. the *EZW* algorithm of Shapiro (1993)) and theoretical analyses (Cohen *et al.*, 1997; Mallat and Falzon, 1998) have shown the power of approximation within a wavelet basis. An alternative is to search large libraries of orthonormal bases, based for example on binary subband coding trees. This leads to wavelet packets (Coifman and Wickerhauser, 1992) and rate-distortion optimal solutions (Ramchandran and Vetterli, 1993).

The third question is more complex than it looks at first sight. If there was no cost associated with describing the index set, then clearly I should be the set $\{n\}$ such that

$$| < \varphi_n, f >_{n \in I} | \geq | < \varphi_m, f > |_{m \notin I} \tag{2.5}$$

But when the rate for I is accounted for, it might be more efficient to use a fixed set I for a class of objects. For example, in the jointly Gaussian case, the optimal procedure chooses a fixed set of Karhunen-Loève basis vectors (namely those corresponding to the largest eigenvalues) and spends all the rate to describe the coefficients with respect to these vectors. Note that a fixed subset corresponds to a linear approximation procedure (before quantization, which is itself non-linear) while choosing a subset as in (2.5) is a non-linear approximation method.

It is easy to come up with examples of objects for which non-linear approximation is far superior to linear approximation. Consider a step function on $[0, 1]$, where the step location is uniformly distributed on $[0, 1]$. Take the Haar wavelet basis as an orthonormal basis for $[0, 1]$. It can be verified that the approximation error using M terms is of the order

$$\epsilon_L \sim 1/M \tag{2.6}$$

for the linear case, while it is of the order

$$\epsilon_{NL} \sim 2^{-M} \tag{2.7}$$

for non-linear approximation using the M largest terms. However, this is only the first part of the rate distortion story, since we still have to describe the M chosen terms.

This rate distortion analysis takes into account that a certain number of scales J have to be represented, and at each scale, the coefficients require a certain number of bits. This split leads to a number of scales $J \sim \sqrt{R}$. The error is the sum of errors of each scale, each of which is of the order $2^{-R/J}$. Together, we get:

$$D_{NL}(R) \sim \sqrt{R} 2^{-\sqrt{R}} \tag{2.8}$$

The quantization question is relatively simple if each coefficient α_n is quantized by itself (so-called scalar quantization). Quantizing several coefficients together (or vector quantization) improves the performance, but increases complexity. Usually, if a "good" basis is used and complexity is an issue, scalar quantization is the preferred method.

The fourth question is a critical one. While approximation in orthonormal bases are very popular, they cannot be the end of the story. Just as not every stochastic process is Gaussian, not all objects will be well represented in an orthonormal basis. In other words, fitting a linear subspace to arbitrary objects is not always a good approximation. But even for objects where basis approximation does well, some other approximation method might do much better. In our step function example studied earlier, a simple minded coding of the step location and the step value leads to a rate-distortion behavior

$$D'(R) \sim 2^{-R/2} \tag{2.9}$$

In the remainder of this paper we are going to study in more detail the difference between wavelet and direct approximation of piecewise polynomial signals.

3. R/D upper bounds for a piecewise polynomial function

Consider a continuous time signal $s(t)$, $t \in [a, b]$, composed of M polynomial pieces; assume that the maximum degree of any polynomial piece is less than or equal to N and that each piece (and therefore the entire signal) is bounded in magnitude by some constant A. The signal is uniquely determined by the M polynomials and by $M-1$ internal breakpoints; by augmenting the set of breakpoints with the interval extremes a and b, we can write:

$$s(t) = \sum_{n=0}^{N} p_n^{(i)} t^n = p_i(t) \text{ for } t_i \le t < t_{i+1} \tag{3.1}$$

where $a = t_0 < t_1 < \ldots < t_{M-1} < t_M = b$ are the breakpoints and the $p_n^{(i)}$ are the i-th polynomial coefficients (with $p_n^{(i)} = 0$ for n larger than the polynomial degree); let $T = (b-a)$.

(a) Polynomial Approximation

In this section we will derive an upper bound on the rate distortion characteristic for a quantized, piecewise polynomial approximation of $s(t)$. For the time being, assume that the values for M, for the degrees of the polynomial pieces, and for the internal breakpoints are provided with arbitrary accuracy by an oracle. The derivation of the operational R/D bound will be carried out in three steps: first we will determine a general R/D upper bound for the single polynomial pieces; secondly, we will determine an R/D upper bound for encoding the breakpoint values; finally, we will determine the jointly optimal bit allocation for the whole signal.

(i) Encoding of one polynomial piece

Consider the i-th polynomial of degree N_i, defined over the support $I_i = [t_i, t_{i+1}]$ of width S_i. Using a local Legendre expansion (see Appendix) we can write (the subscript i is dropped for clarity throughout this section):

$$p(t) = \sum_{n=0}^{N} p_n t^n = \sum_{n=0}^{N} \frac{2n+1}{S} l_n L_I(n;t) \tag{3.2}$$

where $L_I(n;t)$ is the n-th degree Legendre polynomial over I; due to the properties of the expansion it is

$$|l_n| \leq AS \tag{3.3}$$

for all n. The squared error after quantizing the coefficients can be expressed as

$$\begin{aligned} e^2 &= \sum_{n=0}^{N} \left(\frac{2n+1}{S}\right)^2 (l_n - \hat{l}_n)^2 \int_{t_i}^{t_{i+1}} L_I^2(n;t)\, dt = \\ &= S^{-1} \sum_{n=0}^{N} (2n+1)(l_n - \hat{l}_n)^2 \end{aligned} \tag{3.4}$$

where $\hat{l}_n$ are the quantized values. Assume using for each coefficient a different b_n-bit uniform quantizer over the range specified by (3.3) for a step size of $2AS2^{-b_n}$; the total squared error can be upper bounded as:

$$e^2 \leq D_p = A^2 S \sum_{n=0}^{N} (2n+1) 2^{-2b_n} \tag{3.5}$$

For a global bit budget of R_p bits, the optimal allocation is found by solving the following reverse waterfilling problem

$$\begin{cases} \dfrac{\partial D_p}{\partial b_n} = \text{const.} \\ \sum b_n = R_p \end{cases} \tag{3.6}$$

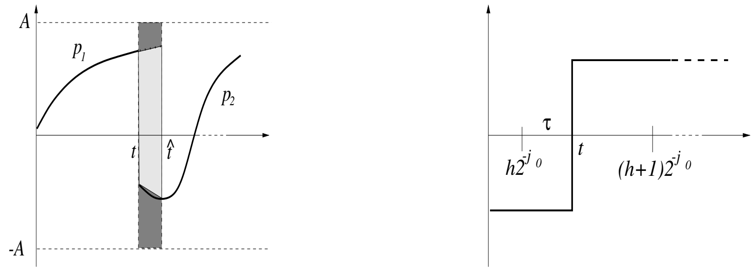
which yields

$$b_n = \frac{R_p}{N+1} + \log_2 \sqrt{\frac{2n+1}{\bar{C}}} \tag{3.7}$$

with

$$\bar{C} = \left[\prod_{n=0}^{N} (2n+1)\right]^{\frac{1}{N+1}}; \tag{3.8}$$

since the geometric mean is always less than or equal to the arithmetic mean we have $\bar{C} \leq (N+1)$, and we finally obtain the following upper bound for the i-th



(a) (b)

Figure 1. Encoding of switchpoints: (a) true error (light area) and general upper bound (dark area); (b) location of the jump at a given wavelet scale.

polynomial piece:

$$D_p(R_p) \leq A^2 S(N+1)^2\, 2^{-\frac{2}{N+1}R_p}. \tag{3.9}$$

(ii) Encoding of switchpoints

Assume that the $M+1$ switchpoints t_i, as provided by the oracle, are quantized with a uniform quantizer over the entire support of the signal. In terms of the overall mean squared error, the error relative to each quantized switchpoint can be upper bounded by (see Figure 1-(a)):

$$e_{t_i}^2 \leq 4A^2\, |t_i - \hat{t}_i|. \tag{3.10}$$

Again, the magnitude of the error is at most one half of the quantizer's step size, so that for a given switchpoint we have:

$$D_t(R_t) \leq 2A^2 T\, 2^{-R_t} \tag{3.11}$$

where R_t is the quantizer's rate.

(iii) Composite R/D bound

The global distortion bound for $s(t)$ is obtained additively as

$$D \leq \sum_{i=1}^{M} D_{p_i}(R_{p_i}) + \sum_{i=0}^{M+1} D_{t_i}(R_{t_i}) \tag{3.12}$$

where $D_{p_i}(R_{p_i})$ and $D_{t_i}(R_{t_i})$ are the bounds in (3.9) and (3.11) respectively, and where the subscript denotes the index of the polynomial pieces.

In order to obtain the optimal bit allocation for the composite polynomial function given an overall rate, it would be necessary to find the constant-slope operating points for all the summation terms in (3.12), as shown in (3.6); the resulting formulas, however, would be entirely impractical due to their dependence on all the polynomial parameters across the whole function. Instead, we choose to derive a coarser but general upper bound by introducing the following simplifications:

- all polynomial pieces are assumed of maximum degree N; this implies that, for polynomials of lower degree, bits are allocated to the zero coefficients as well;
- the support of each polynomial piece S_i is "approximated" by T, the entire function's support; together with the previous assumption, this means that the

waterfilling algorithm will assign the same number of bits R_p to each polynomial piece;

- the origin of the function support (a) is either known or irrelevant; this reduces the number of encoded switchpoints to M;
- all switchpoints are encoded at the same rate R_t.

With these simplifications the rate distortion bound becomes:

$$D(R) \leq A^2 T M \left(2^{-R_t+1} + (N+1)^2\, 2^{-\frac{2}{N+1} R_p}\right) \tag{3.13}$$

where the total bit rate is $R = M(R_t + R_p)$. By the usual reverse waterfilling argument we obtain the optimal allocation:

$$R_p = \frac{N+1}{N+3}\frac{R}{M} + \log_2 K \tag{3.14}$$

$$R_t = \frac{2}{N+3}\frac{R}{M} - \log_2 K \tag{3.15}$$

with $K = (2N+2)^{(N+1)/(N+3)}$. Using the relation (for $N > 0$)

$$2K + (N+1)^2 K^{-\frac{2}{N+1}} \leq 2(N+1)^2 \tag{3.16}$$

a simplified global upper bound is finally:

$$D_P(R) \leq 2A^2 T\, M (N+1)^2\, 2^{-\frac{2}{N+3}\frac{R}{M}}. \tag{3.17}$$

(*b*) *Wavelet-based Approximation*

In this section we will obtain an upper bound for the case of a quantized nonlinear approximation of $s(t)$ using a wavelet basis over $[a, b]$. The derivation follows the lines in the manuscript by Cohen *et al.* (1997) and assumes the use of compact support wavelets with at least $N+1$ vanishing moments (Cohen *et al.*, 1993).

(i) Distortion

If the wavelet has $N+1$ vanishing moments, then the only nonzero coefficients in the expansion correspond to wavelets straddling one or more switchpoints; since the wavelet has a compact support as well, each switchpoint affects only a finite number of wavelets at each scale, which is equal to the length of the support itself. For $N+1$ vanishing moments, the wavelet support L is

$$L \geq 2N+1 \tag{3.18}$$

and therefore, at each scale j in the decomposition the number of nonzero coefficients C_j is bounded as

$$L \leq C_j \leq ML. \tag{3.19}$$

For a decomposition over a total of J levels, if we neglect the overlaps at each scale corresponding to wavelets straddling more than a single switchpoint we can upper bound the total number of nonzero coefficients C as

$$C \leq MLJ. \tag{3.20}$$

It can be shown (see for example Mallat, 1997) that the nonzero coefficients decay with increasing scale as

$$|c_{j,k}| \leq ATW\, 2^{-j/2} \tag{3.21}$$

where W is the maximum of the wavelet's modulus. Using the same high-resolution b-bit uniform quantizer for all the coefficients† with a stepsize of $2ATW\, 2^{-b}$ we obtain the largest scale before all successive coefficients are quantized to zero:

$$J = 2b - 2. \tag{3.22}$$

With this allocation choice the total distortion bound is $D = D_q + D_t$ where

$$D_q = \sum_k \sum_{j=0}^{J} (c_{j,k} - \hat{c}_{j,k})^2 \tag{3.23}$$

is the quantization error for the coded nonzero coefficients and where

$$D_t = \sum_k \sum_{j=J+1}^{+\infty} c_{j,k}^2 \tag{3.24}$$

is the error due to the wavelet series truncation after scale J (in both summations the index k runs over the nonzero coefficients in each scale). Upper bounding the quantization error in the usual way and using the bound in (3.21) for each discarded coefficient we obtain

$$\begin{aligned} D &\leq C(ATW)^2\, 2^{-2b} + ML\,(ATW)^2\, 2^{-J} = \\ &= ML\,(ATW)^2 \left(1 + \frac{1}{4}J\right) 2^{-J}. \end{aligned} \tag{3.25}$$

(ii) Rate

Along with the quantized nonzero coefficients, we must supply a significance map indicating their position; due to the structure of $s(t)$, 2 bits per coefficients suffice to indicate which of the next-scale wavelet siblings (left, right, both, or none) are nonzero. The total rate therefore is

$$R = C(b+2) \leq MLJ(J/2+3) \tag{3.26}$$

where we have used (3.20) and (3.22). In our high resolution hypothesis it is surely going to be $b \geq 4$ and therefore we can approximate (3.26) as

$$R \leq MLJ^2. \tag{3.27}$$

(iii) Global upper bound

Eqn. (3.25) provides a distortion bound as a function of J; in turn, J is a function of the overall rate as in (3.27). Combining the results we obtain the overall rate/distortion bound:

$$D_W(R) \leq (ATW)^2\, M(2N+1) \left(1 + \frac{1}{4}\sqrt{\frac{1}{2N+1}\frac{R}{M}}\right) 2^{-\sqrt{\frac{1}{2N+1}\frac{R}{M}}} \tag{3.28}$$

† In their paper, Cohen and coauthors (1997) perform a more detailed analysis in which the quantizer's stepsize is varied according to the decay in (3.21); this however affects only the constants in the R/D bound and not the asymptotics.

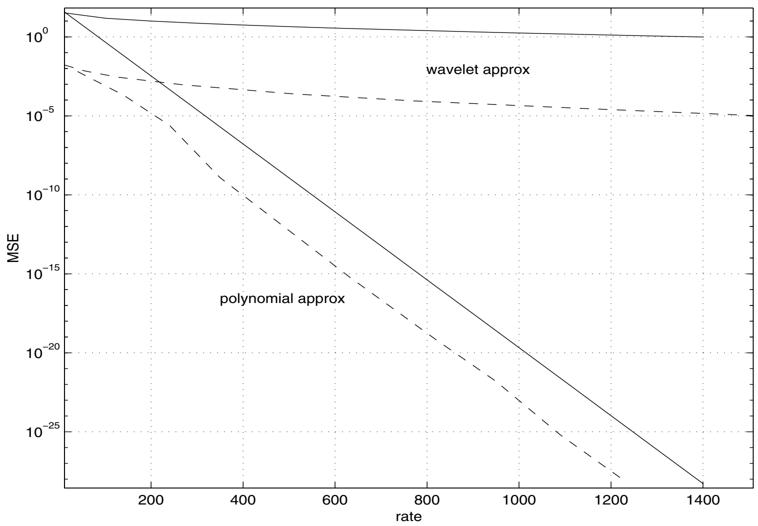


Figure 2. Theoretical (solid) and experimental (dashed) R/D curves.

where we have assumed a minimum support wavelet, for which $L = 2N + 1$.

(*c*) *Commentary*

To recapitulate, the two upper bounds obtained in the previous sections are of the form:

$$\begin{array}{ll} \textit{polynomial approximation} & D_P(R) = C_p' 2^{-C_p R} \\ \textit{wavelet approximation} & D_W(R) = C_w'(1 + \alpha\sqrt{C_w R})2^{-\sqrt{C_w R}} \end{array}$$

Since these are upper bounds, we are especially concerned with their tightness. Unfortunately, as we have seen, many simplifications have been introduced in the derivation, some of which are definitely rather crude; we will therefore concentrate on the rate of decay of the R/D function rather than on the exact values of the constants. In order to gauge the applicability of the theoretical bounds, we can try to compare them to the actual performance of practical coding systems. A word of caution is however necessary: in order to implement the coding schemes described above, which are derived for continuous time functions, a discretization of the test data is necessary; as a consequence, an implicit granularity of the time axis is introduced, which limits the allowable range for both the breakpoint quantization rate and for the number of decomposition levels in the wavelet transformation. Unfortunately, computational requirements soon limit the resolution of the discretization: in our experiments we have used 2^{16} points. The two approximation techniques have been applied to randomly generated piecewise polynomial functions with parameters $A = 1$, $T = 1$, $N = 4$ and $M = 4$; Daubechies wavelets with 5 vanishing moments on the $[0, 1]$ interval have been used for the decomposition. The results are shown in Figure 2: the solid lines and the dashed lines display the R/D bound and the operational R/D curve respectively for the polynomial and wavelet approximation strategies averaged over 50 function realizations.

A closer inspection of the R/D curves shows that, especially for the wavelet case, there appears to be a large numerical offset between theoretical and practical values even though the rate of decay is correct. This simply indicates that the bounds for the constants in (3.28) are exceedingly large and the question is whether we can arrive at tighter estimates. In the absence of a detailed statistical description for the characteristic parameters of $s(t)$, the answer remains rather elusive in the general case; we can however try to develop our intuition by studying in more detail a toy problem involving minimal-complexity elements and a simple statistical model for the approximated function. The availability of a particular statistical model for the generating process allows us to derive an R/D result in expectation, which is hopefully tighter. Consider the simple case in which $N = 0$, $M = 2$, $T = 1$, and $A = 1/2$: the resulting $s(t)$ is simply a step function over, say, $[0, 1)$; we will assume that the location of the step transition t_0 is uniformly distributed over the support and that the values of the function left and right of the discontinuity are uniformly distributed over $[-1/2, 1/2]$. Having a piecewise constant function allows us to use a Haar wavelet decomposition over the $[0, 1]$ interval; recall that the Haar scaling function and wavelet have a single vanishing moment and they admit the closed-form representation

$$\varphi_0(t) = 1 \tag{3.29}$$

$$\psi_{j,k}(t) = \begin{cases} +2^{j/2} & k2^{-j} \le t < (k+1/2)2^{-j} \\ -2^{j/2} & (k+1/2)2^{-j} \le t < (k+1)2^{-j} \\ 0 & \text{elsewhere} \end{cases} \tag{3.30}$$

from which it is easy to see that there is no overlap between wavelets within a scale. Now the following facts hold:

- because of the absence of overlap, at each scale we have exactly one nonzero coefficient†; the relation in (3.20) becomes exact:

$$C = J; \tag{3.31}$$

- under the high resolution hypothesis for a b-bit quantizer, the quantization error becomes an uniform random variable over an interval half a step wide; the expected error for each quantized coefficient is therefore $2^{-2b}/12$;
- the series truncation error (3.24) is, in expectation,

$$E[D_t] = (1/36)\, 2^{-(J+1)}; \tag{3.32}$$

(for a proof, see Appendix B);

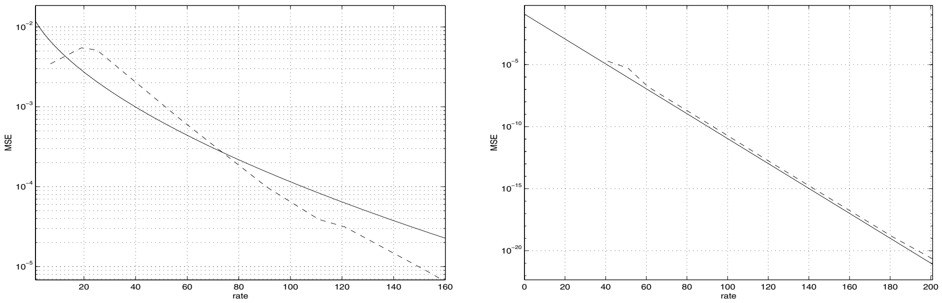
- again, due to the non overlapping properties of the Haar wavelet, we can rewrite (3.27) simply as $R \le J^2$.

With these values, the R/D curve, *in expectation*, becomes:

$$D_W(R) \le \frac{1}{72}\left(1 + \frac{3}{4}\sqrt{R}\right)\, 2^{\sqrt{R}}). \tag{3.33}$$

Figure 3-(a) displays this curve along with the experimental results (dashed line); we can now see that the numerical values agree to within the same order of magnitude. For completeness, the expected R/D behavior for the polynomial

† In fact, a further consequence of the non-overlapping wavelets is that we need only one bit per coefficient to encode the significance map; for simplicity we will however use the general rate requirement (3.26) both in the theoretical derivation and in the algorithmic implementation.



(a) (b)

Figure 3. Theoretical (solid) and experimental (dashed) R/D curves for the step function approximation: (a) Haar wavelet approximation, (b) polynomial approximation.

approximation of the above step function (obtained with a similar, simplified analysis) turns out to be:

$$D_P(R) = \frac{1}{6\sqrt{2}}\,2^{-R/2} \tag{3.34}$$

and the curve, together with its experimaental counterpart, is displayed in Figure 3-(b).

4. R/D optimal approximation

We have seen that the direct polynomial approximation displays a far better rate-distortion asymptotic behavior than the standard nonlinear wavelet approximation. However, the polynomial bound was derived under two special hypotheses which are not generally met in practice: the availability of an “oracle” and the use of high resolution quantizers. Since the goal of most approximation techniques is a parsimonious representation of the data for compression purposes, the question arises naturally: what is the best coding strategy in a practical setting where the polynomial parameters are initially unknown and the bit rate is severely constrained? The problem can be cast in an *operational* rate-distortion framework where the objective is to “fit” different polynomial pieces to the function subject to a constraint on the amount of resources used to describe the modelization and where the goodness of fit is measured by a global cost functional such as the MSE. In this constrained allocation approach, both breakpoints and local models must be determined jointly, since the optimal segmentation is a function of the family of approximants we allow for and of the global bitrate. In particular, for low rates, the available resources might not allow for a faithful encoding of all pieces and a globally optimal compromise solution must be sought for, possibly by lumping several contiguos pieces into one or by approximating the pieces by low-degree polynomials which have lighter description complexity.

In the following, we will illustrate a “practical’ algorithm which addresses and solves these issues, and whose performance matches and extends to the low bitrate case the oracle-based polynomial modeling. Please note that now we are entering an algorithmic scenario where we perforce deal with discrete-time data vectors rather than continuous-time functions; similarly to the experimental results of the previous section, granularity of the involved quantities and computational requirements are now important factors.

(*a*) *Joint segmentation and allocation*

Consider an K-point data vector $\boldsymbol{x} = x_1^K$, which is a sampled version of a piecewise polynomial function $s(t)$ over a given support interval. The goal is to fit the data with local polynomial models as to minimize the global mean squared error of the approximation under a given rate constraint. This defines an *operational* rate-distortion curve which is tied to the family of approximation models we choose to use. This initial choice is the crucial "engineering" decision of the problem and is ruled by a-priori knowledge on the input data (polynomial pieces of maximum degree N) and by economical considerations in terms of computational requirements. In particular, we choose a fixed, limited set of possible rates associated to a polynomial model of given degree, with quantization of the individual coefficients following the line of (3.7). The validity of such design parameters can only be assessed via the performance measure yielded by the operational R/D curve.

In the following we will assume a family of Q polynomial models, which is the aggregate set of polynomial prototypes from degree 0 to N with different quantization schemes for the parameters (more details later). For the data vector $\boldsymbol{x}$ define a *segmentation* $\boldsymbol{t}$ as a collection of $n+1$ time indices: $\boldsymbol{t} = \{t_0 = 1 < t_1 < t_2 < \ldots < t_{n-1} < t_n = K+1\}$. The number of segments defined by $\boldsymbol{t}$ is $\sigma(\boldsymbol{t})$, $1 \le \sigma(\boldsymbol{t}) \le K$, with the i-th segment being $x_{t_i}^{t_{i+1}-1}$; segments are strictly disjoint. Let $T_{[1,K]}$ be the set of all possible segmentations for $\boldsymbol{x}$, which we will simply write as T when the signal range is self-evident; it is clearly $|T_{[1,K]}| = 2^{K-1}$. Parallel to a segmentation $\boldsymbol{t}$, define an *allocation* $\boldsymbol{w}(\boldsymbol{t})$ as a collection of $\sigma(\boldsymbol{t})$ model indices w_i, $1 \le w_i \le Q$; let $W(\boldsymbol{t})$ be the set of all possible allocations for $\boldsymbol{t}$, with $|W(\boldsymbol{t})| = Q^{\sigma(\boldsymbol{t})}$. Again, when the dependence on the underlying segmentation is clear, we will simply write $\boldsymbol{w}$ instead of $\boldsymbol{w}(\boldsymbol{t})$.

For a given segmentation $\boldsymbol{t}$ and a related allocation $\boldsymbol{w}$, define $R(\boldsymbol{t}, \boldsymbol{w})$ as the cost, in bits, associated to the sequence of $\sigma(\boldsymbol{t})$ polynomial models and define $D(\boldsymbol{t}, \boldsymbol{w})$ as the cumulative squared error of the approximation. Since there is no overlap between segments and the polynomial models are applied independently, we can write

$$D(\boldsymbol{t}, \boldsymbol{p}) = \sum_{i=1}^{\sigma(\boldsymbol{t})} d^2(\hat{\boldsymbol{p}}(w_i); t_i, t_{i+1}); \tag{4.1}$$

more in detail,

$$d^2(\hat{\boldsymbol{p}}(w_i); t_i, t_{i+1}) = \|\hat{\boldsymbol{p}}(w_i) V_{(t_{i+1}-t_i)} - x_{t_i}^{t_{i+1}-1}\|^2 \tag{4.2}$$

where V is a Vandermonde matrix of size $N \times (t_{i+1} - t_i)$ and where $\hat{\boldsymbol{p}}(w_i)$ is an $N+1$-element vector containing the estimated polynomial coefficients for the i-th segment quantized according to model w_i (the high order coefficients being zero for model orders less than N). We will assume that the polynomial coefficients are coded independently and that their cost in bits is a function $b(\cdot)$ of the model's index only. An important remark at this point is that, by allowing for a data-dependent segmentation, *information about the segmentation and the allocation* *themselves must be provided along with the polynomial coefficients.* This takes the form of side information, which uses up part of the global bit budget of the R/D optimization and must therefore be included in the expression for the overall rate. We will then write

$$R(\boldsymbol{t}, \boldsymbol{w}) = \sum_{k=1}^{\sigma(\boldsymbol{t})} (c + b(w_k)) = \sum_{k=1}^{\sigma(\boldsymbol{t})} r(p_k) \tag{4.3}$$

where c is the side information associated to a new segment specifying its length and the relative polynomial order and quantization choice†.

Our goal is to arrive at a minimization of the global squared error with respect to the local polynomials and to the data segmentation using the global rate as a parameter controlling the number of segments and the distribution of bits amongst the segments. Formally, this amounts to solving the following constrained problem:

$$\begin{cases} \min\limits_{\boldsymbol{t} \in T} \min\limits_{\boldsymbol{w} \in W(\boldsymbol{t})} \{D(\boldsymbol{t}, \boldsymbol{w})\} \\ \\ R(\boldsymbol{t}, \boldsymbol{w}) \le R_C. \end{cases} \tag{4.4}$$

While at first the task of minimizing (4.4) seems daunting, requiring $O(Q^N)$ explicit comparisons, we will show how it can be solved in polynomial time for almost all rates using standard optimization techniques.

(*b*) *Efficient Solution*

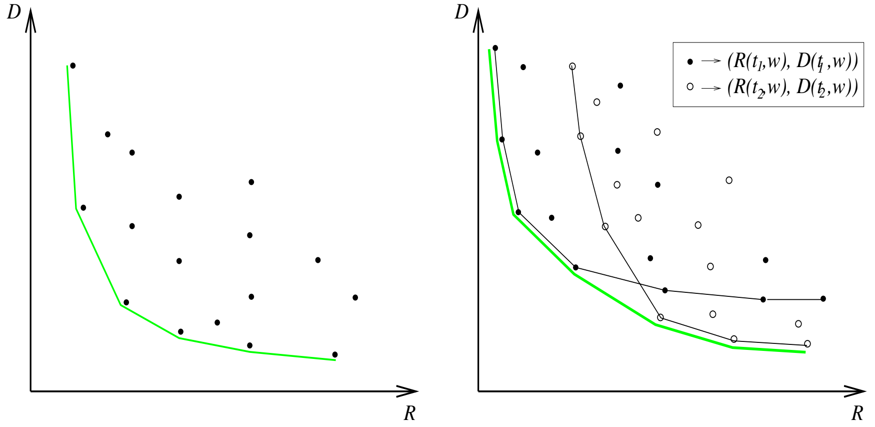
The problem of optimal resource allocation has been thoroughly studied in the context of quantization and coding for discrete datasets (Gersho and Gray, 1992) and has been successfully applied to the context of signal compression and analysis (Ramchandran and Vetterli, 1993; Xiong *et al.*, 1994; Prandoni *et al.*, 1997). In the following we will rely extensively on the results by Shoham and Gersho (1988), to which the reader is referred for details and proofs.

For the time being assume that a segmentation $\boldsymbol{t}_0$ is given (a fixed-window segmentation, for instance) and that the only problem is to find the optimal allocation of polynomial pieces; each allocation defines an operational point in the R/D plane as in Figure 4-(a) and the inner minimization in (4.4) requires us to find the allocation yielding the minimum distortion amongst all the allocations with the same given rate. However, if we restrict our search to the convex hull of the entire set of R/D points, the minimization can be reformulated using Lagrange multipliers: define a functional $J(\lambda) = D(\boldsymbol{t}, \boldsymbol{w}) + \lambda R(\boldsymbol{t}, \boldsymbol{w})$; if, for a given λ,

$$\boldsymbol{w}^* = \arg \min_{\boldsymbol{w} \in W(\boldsymbol{t}_0)} \{J(\lambda)\} \tag{4.5}$$

then $\boldsymbol{w}^*$ (star superscripts denote optimality) defines a point on the convex hull

† Here, the cost of side information is assumed constant for simplicity. However, no major changes in the subsequent derivation are needed if this cost depends on the segment's parameters.



(a)

(b)

Figure 4. R/D convex hulls: (a) convex hull for a single segmentation; (b) composite convex hull for two segmentations.

which solves the problem:

$$\begin{cases} \min\limits_{\boldsymbol{w}\in W(\boldsymbol{t}_0)} \{D(\boldsymbol{t}_0, \boldsymbol{w})\} \\ \\ R(\boldsymbol{t}, \boldsymbol{w}) \leq R(\boldsymbol{t}_0, \boldsymbol{w}^*). \end{cases} \tag{4.6}$$

If we now let the segmentation vary, we simply obtain a larger population of operational R/D points which are indexed by segmentation-allocation pairs as in Figure 4-(b). Again, if we choose to restrict the minimization to the convex hull of the composite set of points, we can solve the associated Lagrangian problem as a double minimization:

$$J^*(\lambda) = \min_{\boldsymbol{t}\in T} \min_{\boldsymbol{w}\in W(\boldsymbol{t})} \{J(\lambda)\}; \tag{4.7}$$

it should be noted that the restriction to the convex hull is of little practical limitation when the set of R/D points sufficiently dense; this is indeed the case given the cardinalities of T and W.

Even in the form of (4.7) the double minimization would still require an exhaustive search over all R/D points, in addition to a search for the optimal λ. By taking the structure of rate and error into account, we can however rewrite (4.7) as:

$$J^*(\lambda) = \min_{\boldsymbol{t}\in T} \min_{\boldsymbol{w}\in W(\boldsymbol{t})} \{\sum_{k=1}^{\sigma(\boldsymbol{t})} (d^2(\hat{\boldsymbol{p}}(w_k); t_k, t_{k+1}) + \lambda r(w_k))\} \tag{4.8}$$

Since all quantities are nonnegative and the segments are non overlapping, the inner minimization over $W(\boldsymbol{t})$ can be carried out independently term-by-term, reducing the number of comparisons to $Q\sigma(\boldsymbol{t})$ per segmentation. Now the key observation is that, whatever the segmentation, all segments are coded with the same rate/distortion tradeoff as determined by λ; therefore, for a given λ, we can determine the optimal $\boldsymbol{t}$ (in the sense of (4.4)) using dynamic programming (Bellman, 1957). Indeed, suppose a breakpoint t belongs to $\boldsymbol{t}^*$, the optimal segmentation; then it is easy to see that

$$J^*_{[1,N]}(\lambda) = J^*_{[1,t]}(\lambda) + \min_{\boldsymbol{t}\in T_{[t,N]}} \min_{\boldsymbol{w}\in W(\boldsymbol{t})} \{J(\lambda)\} \tag{4.9}$$

(where subscripts indicate the signal range for the minimization). In other words, if t is an optimal breakpoint, the optimal cost functional for x_1^{t-1} is independent of subsequent data. This defines an incremental way to jointly determine the optimal segmentation and allocation as a recursive optimality hypothesis for all data points: for $0 \leq t \leq N$,

$$J^*_{[1,t]}(\lambda) = \min_{1\leq\tau\leq t-1} \{J^*_{[1,\tau]}(\lambda) + \min_{1\leq w\leq Q} \{d^2(\hat{\boldsymbol{p}}(w);\tau,t) + \lambda r(w)\}\} \tag{4.10}$$

(where $J^*_{[1,0]}(\lambda) = 0$). At each step t, only the new $J^*_{[1,t]}(\lambda)$ and the minimizing τ need be stored. The total number of comparisons for the double minimization is therefore $O(K^2)$. The latter must be iterated over λ until the rate constraint in (4.4) is met; luckily, the overall rate is a monotonically non increasing function of λ (for a proof, see again the work by Shoham and Gersho (1988)) so that the optimal value can be found with a fast bisection search (see Ramchandran and Vetterli, 1993).

(*c*) *Implementation and results*

In the implementation of the dynamic segmentation algorithm we have chosen a simplified set of quantization schemes. At low bitrates, Equation (3.7) states that the optimal bit distribution for a set of polynomial coefficients is basically uniform. We choose four possible allocations of 4, 8, 12, and 16 bits for the single coefficient, with the total bitrate of a polynomial piece linearly dependent on its degree. A Least Squares problem is solved for all orders from zero to N for each possible segment in an incremental fashion paralleling (4.10); this involves extending the QR factors of an order-N Vandermonde matrix by a new point at each step, which can be performed efficiently by means of Givens rotations. Side information for each segment is composed of two bits to signal the quantization scheme, $\lceil \log_2 N \rceil$ bits for the order of the polynomial model and $\lceil \log_2 K \rceil$ bits for the length of the segment. Finally, the computation in (4.10) can be efficiently organized on a trellis, where intermediate data are stored prior to the iteration over λ; further algorithmic details, omitted here, can be found in a related papers by the present authors (1999). The final computational requirements for the global minimization are on the order of $O(K^3)$, with storage on the order of $O(K^2)$.

We can now compare the experimental results of the optimal allocation algorithm with the polynomial approximation R/D bound obtained using an oracle; however, since here the interest lies in very low bit rates as well, we need to somehow refine the bound in (3.17). In fact, under severe rate constraints, there might not be enough bits to encode the exact structure of the function and the dynamic algorithm will be forced to use a coarse segmentation in which several contiguous polynomial pieces are approximated by just one model; in the limit, when the rate goes to zero, the approximation error approaches the integral of $s^2(t)$ over the entire support of the function. This is not reflected by Equation (3.17), where the expression for the error always assumes M distinct pieces. By approximating the maximum error by $4A^2T$, we can define a more appropriate R/D bound for

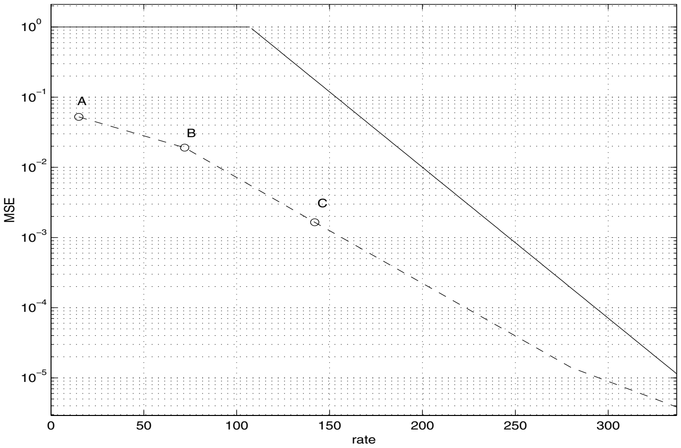


Figure 5. Theoretical (solid) and experimental (dashed) R/D curves for the dynamic segmentation algorithm.

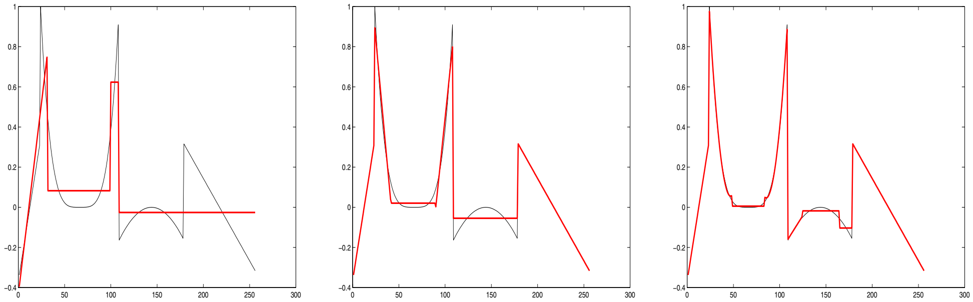


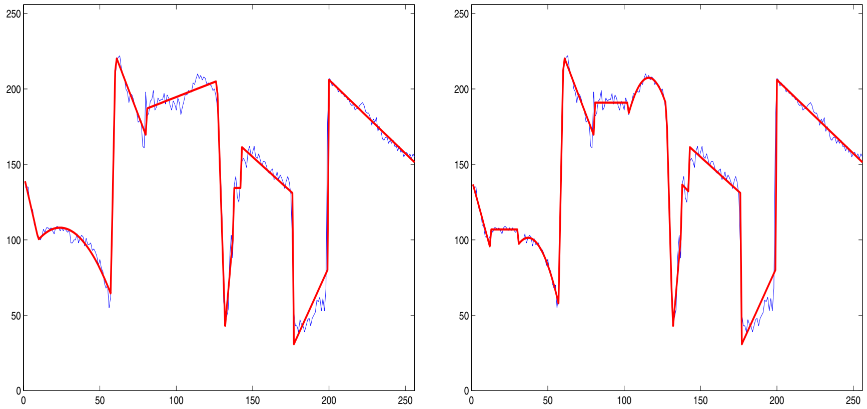
Figure 6. Approximations provided by the dynamic segmentation algorithm corresponding to points A, B, and C on the R/D curve in Figure 5.

the polynomial case as

$$D'_P(R) = \min\{4A^2T, D_P(R)\}. \qquad (4.11)$$

Figure 5 shows the numerical results obtained for a set of piecewise polynomial functions as in the previous experiment; the underlying sampling is however coarser here ($K = 2^8$), due to the heavier computational load. As before, the solid line indicates the new theoretical upper bound and the dashed line the R/D performance of the dynamic algorithm. It is also interesting to look more in detail at the segmentation/allocation choices performed by the algorithm for different bitrate constraints; this is displayed in Figures 6-(a),(b), and (c) with respect to the R/D points A, B, and C marked by a circle in Figure 5. In Figure 6 the thin line shows the original piecewise polynomial function while the thick lines shows the algorithmic results; while not always very intuitive, these low bit rate approximations are nonetheless optimal in a MSE sense.

As a final note, we can ask ourselves two further questions: how does this framework extend to real-world signals, which are clearly not exactly piecewise polynomial ? And more, can this framework be applied and compared to practical coding scenarios in which wavelets are known to perform very well, such as image



(a) (b)

Figure 7. Piecewise polynomial approximations of a line of Lena.

compression ? Unfortunately, dynamic programming techniques do not work for two-dimensional problems, and it is not clear how to fit polynomial surfaces in a globally optimal way. Yet, we can gain some intuition about both questions by looking at Figures 7-(a) and (b). The thin line represents a single line of the "Lena" image, for a total of 256 pixels; the thick lines are the piecewise polynomial approximations of the data, at increasing rates, obtained with the dynamic segmentation algorithm introduced above. We could argue that, for increasing bitrates, the algorithm captures more and more finely the local polynomial trends underlying the image surfaces, while the finer details can be represented as an additive, noise-like residual. Whether this can lead to an efficient approximation scheme for images is however hard to say at present.

Appendix A. Local Legendre Expansion

Legendre polynomials are usually defined over the $[-1,1]$ interval by the recurrence relation

$$(n+1)L(n+1;t) = (2n+1)t\,L(n;t) - nL(n-1;t) \tag{A 1}$$

where $L(n;t)$ is the Legendre polynomial of degree n. They constitute an orthogonal basis for $L[-1,1]$:

$$\int_{-1}^{1} L(n;t)L(m;t)\,dt = \frac{2}{2n+1}\delta(n-m) \tag{A 2}$$

and in particular, for a polynomial $p(t)$ of degree N over $[-1,1]$, we can write

$$l_n = \int_{-1}^{1} L(n;t)p(t)\,dt,\ n = 0,\dots,N \tag{A 3}$$

$$p(t) = \sum_{n=0}^{N} \frac{2n+1}{2} l_n L(n;t). \tag{A 4}$$

Since $|L(n;t)| \leq 1$ for all n, $|l_n| \leq 2\sup_{[-1,1]}[p(t)]$.

A *local* Legendre expansion over the interval $I = [\alpha,\beta]$ can be obtained by defining a translated set of orthogonal polynomials

$$L_I(n;t) = L(n; \frac{2}{\beta-\alpha}t - \frac{\alpha+\beta}{\beta-\alpha}); \tag{A 5}$$

the orthogonality relation becomes

$$\int_\alpha^\beta L_I(n;t)L_I(m;t)\,dt = \frac{\beta-\alpha}{2n+1}\delta(n-m) \tag{A 6}$$

and the analysis/synthesis formulas can be written as:

$$l_n = \int_\alpha^\beta L_I(n;t)p(t)\,dt,\ n = 0,\dots,N \tag{A 7}$$

$$p(t) = \sum_{n=0}^{N} \frac{2n+1}{\beta-\alpha} l_n L_I(n;t). \tag{A 8}$$

Appendix B. Estimate of the series truncation error

The estimate in (3.32) can be obtained as follows: assume that at scale j_0 the step discontinuity at t_0 falls within the interval $[h2^{-j_0}, (h+1)2^{-j_0})$ for some h. Then in the Haar wavelet series for $s(t)$ the indices of the nonzero coefficients $c_{j,k}$ for $j > j_0$ satisfy

$$h2^{j-j_0} \leq k < (h+1)2^{j-j_0}. \tag{B 1}$$

The wavelet set $\{\psi_{j,k}(t)\}$ with j and k as above form an orthonormal basis for the $[h2^{-j_0}, (h+1)2^{-j_0})$ interval minus the addition of a scaling function $\varphi(t) = 2^{j_0/2}$ over the same interval. We can therefore write:

$$\begin{aligned} D_t = \sum_{j=0}^{\infty} \sum_{k=h2^{j-j_0}}^{(h+1)2^{j-j_0}-1} c_{j,k}^2 = \\ = \int_{h2^{-j_0}}^{(h+1)2^{j-j_0}} s^2(t)\,dt - \left[2^{-j_0/2} \int_{h2^{-j_0}}^{(h+1)2^{j-j_0}} s(t)\,dt\right]^2 \end{aligned} \tag{B 2}$$

where we have used Parseval's identity.

Now consider the location of the step (see Figure 1-(b)); let $\tau = t_0 - h2^{-j_0}$ be the distance between the discontinuity and the origin of the interval. Due to the properties of $s(t)$ we can safely assume that τ is uniformly distributed over $[0, 2^{-j_0}]$. We can now write

$$\begin{aligned} D_t = \tau x_1^2 + (2^{-j_0} - \tau)x_2^2 - \\ 2^{j_0}(\tau^2 x_1^2 + (2^{-j_0} - \tau)^2)x_2^2 + 2\tau(2^{-j_0} - \tau)x_1 x_2) \end{aligned} \tag{B 3}$$

where $x_{1,2}$ are the values of $s(t)$ left and right of the jump, respectively. Taking expectations over the independent quantities τ, x_1, and x_2, where $x_{1,2} \in \mathcal{U}[-1/2, 1/2]$, we finally have:

$$E[D_t] = \frac{1}{36}\,2^{-j_0}. \tag{B 4}$$

References

R. Bellman. *Dynamic Programming.* Princeton University Press, 1957.

T. Berger. *Rate Distortion Theory.* Prentice-Hall, Englewood Cliffs, NJ, 1971.

A. Cohen, I. Daubechies, O. Guleryuz, and M. Orchard. On the importance of combining wavelet-based non-linear approximation in coding strategies. Manuscript, 1997.

A. Cohen, I. Daubechies, and P. Vial. Wavelet bases on the interval and fast algorithms. *J. of Appl. and Comput. Harmonic Analysis*, 1, 1993.

R. R. Coifman and M. V. Wickerhauser. Entropy-based algorithms for best basis selection. *IEEE Tran. on IT*, 38(2):713–718, March 1992.

A. Gersho and R.M. Gray. *Vector Quantization and Signal Compression.* Kluver, 1992.

S. Mallat. *A Wavelet Tour of Signal Processing.* Academic Press, San Diego, CA, 1997.

A. Ortega and K. Ramchandran. Rate-distortion methods for image and video compression. *IEEE Signal Processing Magazine*, Oct. 1998.

P. Prandoni and M. Vetterli. R/D optimal linear prediction. Submitted to *IEEE Trans. ASSP*, 1999.

P. Prandoni, M. Goodwin, and M. Vetterli. Optimal time segmentation for signal modeling and compression. In *Proc. ICASSP*, volume 3, pages 2029–2032, Munich, April 1997.

K. Ramchandran and M. Vetterli. Best wavelet packet bases in a rate-distortion sense. *IEEE Tran. on IP*, 2(2):160–175, April 1993.

E. Riskin. Optimal bit allocation via the G-BFOS algorithm. *IEEE Trans. IT*, 37(2):400–402, March 1991.

C. E. Shannon. A mathematical theory of communication. *Bell Syst. Tech. Journal*, 27, 1948.

C. E. Shannon. Coding theorems for a discrete source with a fidelity criterion. *IRE Nat. Conv. Rec.*, pages 142–163, 1959.

J. M. Shapiro. Embedded image coding using zerotrees of wavelet coefficients. *IEEE Tr. on Signal Processing*, 41(12):3445–3462, Dec 1993.

Y. Shoham and A. Gersho. Efficient bit allocation for an arbitrary set of quantizers. *IEEE Trans. Acoust., Speech, and Signal Proc.*, 36(9):1445–1453, September 1988.

S. Mallat and F. Falzon. Analysis of low bit rate image transform coding. *IEEE Tr. SP*, 46(4):1027–1042, Apr 1998.

Z. Xiong, K. Ramchandran, C. Herley, and M. T. Orchard. Flexible time segmentations for time-varying wavelet packets. *IEEE Proc. Intl. Symp. on Time-Frequency and Time-Scale Analysis*, pages 9–12, October 1994.