

Rate-distortion optimized tree based coding algorithms

Rahul Shukla¹, Pier Luigi Dragotti, Minh N. Do, and Martin Vetterli
Dept. Communication Systems, EPFL, CH-1015 Lausanne, Switzerland

email: {rahul.shukla, pierluigi.dragotti, minh.do, martin.vetterli}@epfl.ch

Abstract — This paper addresses the problem of efficient coding of a simple class of signals, namely piecewise polynomials. For this signal class, we develop a coding algorithm, which achieves oracle like rate-distortion (R-D) behavior in the high bit rate regime and with a reasonable computational complexity. For the 1-D case, our scheme is based on the binary tree segmentation of the signal and an optimal bit allocation strategy among the different signal segments. The scheme further encodes the similar neighbors *jointly* to achieve the right exponentially decaying R-D behavior ($D(R) \sim c_0 2^{-c_1 R}$). We have also shown that the computational cost of the scheme is of the order $O(N \log N)$. We then show that the scheme can be easily extended to the 2-D case, as the quad tree based coding scheme, with the similar R-D behavior and computational cost. Finally, we will conclude with some numerical results.

I. INTRODUCTION

Recently, there has been a growing interest in the study of piecewise polynomial functions as an approximation to piecewise smooth functions. Wavelets have long been considered ideal candidates for piecewise smooth function due to their vanishing moment properties [5]. It was shown in [1, 6] that for piecewise polynomial signals, the distortion of wavelet based coder decays as $D(R) \sim d_0 \sqrt{R} 2^{-d_1 \sqrt{R}}$. However, since such a signal can be precisely described by a finite number of parameters, it is not difficult to realize that R-D behavior of the oracle based method decays as

$$D(R) \sim c_0 2^{-c_1 R} \quad (1)$$

In [6], this R-D behavior has been realized with a polynomial computational cost using dynamic programming. At low rates, such an algorithm works well for piecewise smooth signals too. The basic ingredient is to precisely model singularities, which can also be done in a wavelet scheme using footprints [4]. The dynamic segmentation algorithm in [6] achieves the right R-D characteristic, but its computational cost is high ($O(N^3)$). Moreover, this algorithm cannot be generalized to the 2-D case.

In this paper, we are interested in a coding scheme, which achieves oracle like asymptotic R-D behavior with polynomial complexity for 1-D as well as 2-D signals. In particular, for the 1-D case we present a coding scheme which utilizes binary tree segmentation with optimal bit allocation among different segments. Investigation of the algorithm reveals the inherent weakness in the initial coding scheme, leading to a suboptimal performance ($D(R) \sim c_2 \sqrt{R} 2^{-c_3 \sqrt{R}}$). This leads us to develop an improved coding scheme, which encodes the similar neighbors jointly. In doing so the improved coding scheme achieves the desired R-D behavior given by (1).

More importantly, the optimal binary tree scheme in the 1-D case can be easily extended to the 2-D case as an optimal quadtree scheme with the similar computational complexity. The proposed optimal quadtree scheme also achieves the oracle like R-D performance for some simple classes of images whereas for the 2-D case there is no known algorithm, which achieves the right R-D behavior with a reasonable computational cost. The rest of the paper is organized as follows: first, we will present the classical binary tree algorithm along with the improved binary tree algorithm which achieves the optimal R-D behaviour. Then, we show the extension of 1-D scheme to 2-D using the quadtree based scheme and conclude with some experimental results.

II. 1-D SCENARIO: BINARY TREE ALGORITHMS

Our goal is to implement a coding scheme based on the modeling assumption that signals are piecewise smooth functions. In this case, if we segment the signal into smaller pieces, then each sub-signal can be well represented by a simpler signal (e.g. polynomial) model. For instance, we can choose a simple polynomial model of degree one. Our signal-class of interest can be defined as follows:

Piecewise polynomial signal (PPS): Let $f(t)$ be a piecewise polynomial function defined over the interval $[0, T]$. Assume that $f(t)$ contains N_t internal transition points (singularities). Suppose that the function $f(t)$ is bounded by A and the maximum degree of a polynomial piece is P .

A. The classical binary tree algorithm

The classical tree algorithm utilizes R-D framework with independent coding of the nodes of the tree and an MSE distortion metric. Our scheme employs a binary tree segmentation followed by a coding scheme on each signal block in an operational R-D optimal sense. We employ an operational R-D optimization that is similar to the approach used in [8] in finding the best wavelet packet bases. A decision strategy based on optimizing R-D performance for each signal block is designed so that the coder can decide if a signal block is worth to be further divided and coded with some appropriate quantization level. The algorithm can be summarized as follows [8]:

1. Segmentation of the input signal: a binary tree segmentation scheme is employed.
2. Generation of the optimal R-D curve for each sub-signal assuming the polynomial model.
3. Optimization of this representation to achieve the best reconstructed signal for a given bit rate constraint and distortion measure (MSE). Optimally allocate the bits among the tree leaves using the Lagrangian cost based pruning method for an optimal quality factor(λ) [9].

B. R-D analysis of the classical binary tree scheme

In this section, we will derive an asymptotic R-D behavior of the classical coding algorithm for the piecewise polynomial signal (PPS). Since there are only N_t transition points so at most N_t nodes will have a transition point and rest will be simply

¹This work was supported by Swiss National Science Foundation Grant 20-63664.00.

represented by a polynomial piece without any discontinuity. The optimal tree is obtained via recursively dividing nodes with transition (switch) points. Essentially the binary tree grows only in the region where the algorithm finds transition points. That means the binary tree segmentation algorithm acts like a singularity detector. Suppose that the tree decomposition depth is J , which will be large in the high rate regime. Therefore the total number of terminal-nodes (leaves) in the classical binary tree is bounded as follows:

$$\begin{aligned} N_J &\leq N_{J-1} + N_t \\ \Rightarrow N_J &\leq (J+1)N_t \\ N_J &\sim JN_t, \text{ as } J \text{ is large enough.} \end{aligned} \quad (2)$$

Therefore the number of leaves to be coded grows linearly w.r.t. the decomposition depth J . In the pruned tree, every tree level will have at most $2N_t$ nodes. Therefore the total number of nodes M_J of the pruned tree is bounded as follows

$$\begin{aligned} M_J &\leq 2JN_t + 1 \\ \Rightarrow M_J &\sim 2JN_t \end{aligned} \quad (3)$$

Since there are only N_t transition points, so at most N_t leaves will have a transition point and the remaining JN_t leaves (eq. (2)) will be simply represented by a polynomial piece without any discontinuity. If we simply don't encode the singular leaves, then their distortion will be bounded by A^2T2^{-J} . Let us allocate $\frac{(P+1)}{2}J$ bits to each of the remaining simple leaves. This will make sure that the distortion of every simple leaf (leaf without a singularity) is bounded by $A^2T(P+1)^22^{-J}$ [6]. Therefore both the simple leaves and singular leaves have the distortion of the order $O(2^{-J})$.

Therefore the net distortion is bounded as follows

$$\begin{aligned} D_T &\leq N_t A^2 T 2^{-J} + JN_t A^2 T (P+1)^2 2^{-J} \\ D_T &\sim A^2 T N_t (P+1)^2 J 2^{-J} \end{aligned} \quad (4)$$

The bit rate needed for coding the leaves is $R_{\text{Leaves}} = JN_t \frac{(P+1)}{2} J$. The binary tree split-merge decision variable will consume bits equal to the total number of nodes ($M_J \sim 2JN_t$; from (3)) in the optimal binary tree. Hence the total bit rate is

$$\begin{aligned} R_T &= R_{\text{Tree}} + R_{\text{Leaves}} \sim 2JN_t + JN_t \frac{(P+1)}{2} J \\ \Rightarrow R_T &\sim \frac{(P+1)N_t}{2} J^2, \text{ as } J \text{ is large enough.} \end{aligned} \quad (5)$$

Combining (4) and (5) provides

$$D_T \sim A^2 T N_t (P+1)^2 \sqrt{\frac{2}{(P+1)N_t}} R_T 2^{-\sqrt{\frac{2}{(P+1)N_t}} R_T}.$$

This results into the following theorem.

Theorem 1 *The classical binary tree coding algorithm, which employs the R-D optimization using the parent children pruning, results into the suboptimal asymptotic R-D behavior*

$$D(R) \sim c_0 \sqrt{R} 2^{-c_1 \sqrt{R}} \quad (6)$$

where $c_0 = A^2 T N_t (P+1)^2 \sqrt{\frac{2}{(P+1)N_t}}$ and $c_1 = -\sqrt{\frac{2}{(P+1)N_t}}$, for the piecewise polynomials.

Investigation of the suboptimality: Since the binary tree segmentation acts like a singularity detector, so when the tree grows to capture a singularity in a region, it divides the same polynomial piece repeatedly. The classical coding scheme encodes these subdivided pieces of the same polynomial separately. This independent coding of dependent nodes results in a suboptimal R-D behavior.

C. An improved binary tree coding algorithm

Investigation in the earlier subsection suggests that the inter node dependency should be exploited to achieve the optimal R-D performance. To ensure this, the improved coding scheme employs the same previous classical binary tree coding scheme followed by the neighbor joint coding algorithm, which, in fact, decides whether incoming leaf should be jointly coded with its neighbor or not. Figure 1 illustrates how the improved scheme outperforms the full as well as the classical tree schemes for the piecewise polynomial signal class.

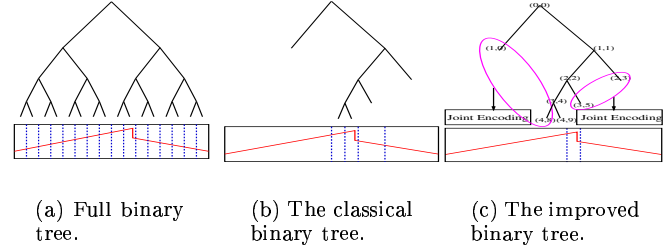


Figure 1: Comparative study of different tree schemes.

D. Neighbor joint coding algorithm

The neighbor joint coding algorithm is as follows:

1. As soon as the neighbor joint coding algorithm receives the leaf information, the algorithm looks for the leaf's spatial neighbor which has already been encountered by the algorithm. Suppose n_i^j represents the j^{th} node at the i^{th} level of the binary tree, then its neighbors ($n_{i_0}^{j_0}$) at the level i_0 are

$$\begin{aligned} \text{Left neighbor : } j_0^L &= 2^{(i_0-i)} j - 1; \\ \text{Right neighbor : } j_0^R &= 2^{(i_0-i)} (j + 1); \end{aligned}$$

In the above formulation, n_0^0 is assumed to be the root node. 2. If the algorithm finds a transmitted neighbor, then it will compare both the leaves using a suitable comparison metric. It has been shown that for R-D optimality, all signal elements (or leaves of the tree) must operate at a constant slope point λ on their R-D curves. This means that the two neighbors will be joined if the sum of the Lagrangian costs ($L(\lambda) = D + \lambda R$) of the neighbors is greater than or equal to the Lagrangian cost of the joint block, i.e., if $(D_{n_1} + \lambda R_{n_1}) + (D_{n_2} + \lambda R_{n_2}) \geq D_{n_j} + \lambda R_{n_j}$. If neighbors are jointly coded then the neighbor joint coding variable will be set to one, otherwise the neighbor joint coding variable will be set to zero and the leaf information will be sent. We need only one bit for the neighbor joint coding variable as the structure of the binary tree always ensures that the number of the immediate neighbors to the incoming leaf is one, so we only have to indicate whether neighbors are encoded jointly or not.

3. If the neighbor joint coding algorithm does not find any transmitted neighbor, then the neighbor joint coding variable will be set to zero and the leaf information will be sent.

E. R-D analysis of the improved binary tree scheme

Lemma 1 *The improved binary tree algorithm, which jointly encodes the similar neighbors, reduces the effective number of the leaves to be encoded to $N_t + 1$, where N_t is the number of the internal transition points in a piecewise polynomial signal.*

Proof: The improved binary tree algorithm employs the neighbor joint coding scheme to encode the similar neighbors jointly. The Lagrangian pruning will join the two neighboring blocks if the joint block does not have the singularity. So there will be at most N_t singular leaves at the decomposition depth J . If J is large enough, then each singularity will lie in a different dyadic leaf. Let us assume that i^{th} singularity lies in the interval $[K_i T 2^{-J}, (K_i + 1) T 2^{-J}]$, where $i = 1, \dots, N_t$

and K_i 's are the integers such that $-1 = K_0 < K_1 < \dots < K_{N_t} < K_{N_t+1} = 2^J$.

If we traverse the binary tree such that we move from left to right in the spatial domain, then all the leaves to the left of the first singularity node ($[K_1 T 2^{-J}, (K_1 + 1) T 2^{-J}]$) at depth J will be joined (pruned) to form the first joint block. Similarly, all the leaves to the right of the first singularity block to the beginning of the second singularity node will be combined to form the second joint block. We can continue this process of joining all the simple leaves (without singularity) between two consecutive singular leaves till we reach the end of the signal. In this way, all the leaves lying in the interval $[(K_{i-1} + 1) T 2^{-J}, K_i T 2^{-J}]$ will be sequentially joined to form one joint leaf. Therefore, the improved tree algorithm results in $N_t + 1$ joint blocks (leaves) and N_t singular leaves. Since the leaves containing the singularity will not be encoded, the number of the encoded leaves becomes $N_t + 1$. Therefore the number of leaves to be coded remains fixed w.r.t. J .

The *high rate analysis* done in Lemma 1 ensures that all the leaves lying in the interval $[(K_{i-1} + 1) T 2^{-J}, K_i T 2^{-J}]$ will be sequentially joined to form one joint leaf. Now let us allocate $\frac{(P+1)}{2} J$ bits to each joint block, so that their distortion is bounded by $A^2 T (P+1)^2 2^{-J}$ [6]. Therefore the bit rate required for coding the leaves is $R_{Leaves} = (N_t + 1) \frac{(P+1)}{2} J$.

In the improved coding scheme, the side information consists of two parts: 1. Bits required to encode the optimal binary tree. 2. Bits required to encode the leaf joint coding tree. The binary tree split-merge decision variable will consume bits equal to the total number of nodes ($M_J \sim 2JN_t$; from (3)) whereas the joint coding decision variable will consume bits equal to the total number of leaves ($N_J \sim JN_t$; from (2)) in the optimal binary tree. Hence the side information costs are: $R_{Tree} \sim 2JN_t$ and $R_{LeavesJointCoding} \sim JN_t$. Hence the total bit rate can be written as follows

$$\begin{aligned} R_T &= R_{Tree} + R_{LeavesJointCoding} + R_{Leaves} \\ \Rightarrow R_T &\sim \frac{(N_t(P+7) + (P+1))}{2} J; \text{ as } J \text{ is large.} \end{aligned} \quad (7)$$

The net distortion is bounded as follows

$$\begin{aligned} D_T &= N_t A^2 T 2^{-J} + (N_t + 1) A^2 T (P+1)^2 2^{-J} \\ \Rightarrow D_T &= (N_t + (N_t + 1)(P+1)^2) A^2 T 2^{-J} \\ \text{Let } c_2 &= (N_t + (N_t + 1)(P+1)^2) A^2 T \\ \Rightarrow D_T &\sim c_2 2^{-\frac{2}{(N_t(P+7) + (P+1))} R_T}; \text{ from (7).} \end{aligned}$$

Therefore the improved tree algorithm achieves the right *asymptotic* R-D behavior, which decays *exponentially*. This result can be stated as Theorem 2.

Theorem 2 *The improved binary tree algorithm, which jointly encodes the similar neighbors, achieves the oracle like exponentially decaying asymptotic R-D behavior*

$$D(R) \sim c_2 2^{-c_3 R} \quad (8)$$

where $c_2 = \frac{(N_t + (N_t + 1)(P+1)^2) A^2 T}{(N_t(P+7) + (P+1))}$ and $c_3 = \frac{2}{(N_t(P+7) + (P+1))}$, for the piecewise polynomials.

F. Computational complexity

Let the size of the signal be N . The classical algorithm has essentially 3 parts:

1. *Generation of the R-D curves* for all the nodes of the full binary tree. Suppose that we are utilizing R_Q different quantizers for $R-D$ function generation. The number of nodes is of the order $O(N)$. Suppose we allow for a tree depth equal to $\log N$, then every level ($i = 0, \dots, \log N$) of the tree contains N pixels, which are divided among 2^i nodes, on which R-D

analysis has been done. Hence the average length of nodes is of the $O(\log N)$. Since R-D computation is proportional to the signal segment length, therefore R-D generation has the computational cost of the order $O(NR_Q \log N)$.

2. *Pruning algorithm* requires the computation of the minimum Lagrangian cost at each node for the chosen quality factor λ . This results in a computational cost of order $O(N \log R_Q)$. The algorithm also performs split merge decision at the nodes which results in a computational complexity of $O(N)$. Hence the pruning algorithm has the computational cost of the order $O(N \log R_Q)$.

3. *Iterative search algorithm for an optimal quality factor* λ^* calls the pruning algorithm for every quality factor (λ) chosen. Suppose we arrive at optimal quality factor (λ^*) in M iterations and in our bisection search algorithm ([8]) $M \sim O(\log N)$. Therefore the search algorithm, for arriving at an optimal quality factor, requires computational complexity of the order $O(N \log R_Q \log N)$.

Therefore the overall computational complexity (C_A) is

$$\begin{aligned} C_A &= O(NR_Q \log N) + O(N \log R_Q \log N) \\ C_A &= O(N(R_Q + \log R_Q) \log N) \simeq O(NR_Q \log N). \end{aligned}$$

Since an optimal binary tree has the number of leaves of the order $O(\log N)$, so the computational complexity of the neighbor joint coding algorithm will be $O(NR_Q \log N)$. Hence the overall computational complexity of the improved coding scheme is also similar to that of the classical coding scheme ($O(NR_Q \log N)$).

III. NUMERICAL RESULTS: 1-D CASE

Numerical experiments are performed for the piecewise polynomial signals defined over the interval $[-1, 1]$. In the experimentation, polynomial coefficients are generated randomly using the uniform distribution on the range $[-1, 1]$. Similarly switch points are also generated randomly using the uniform distribution on the range $[-1, 1]$.

R-D performance comparison is given for the piecewise polynomial signal class with the following characteristics: number of internal switch points $N_t = 32$, signal support is $T = 2$, discretization of the signal support T is done using 2^{17} points, maximum degree of a polynomial piece is two. 16 different quantizers are chosen for the single polynomial coefficient, with total bit rate of a polynomial piece linearly dependent on its degree. Simulation results shown for the *piecewise polynomial signals* in Figure 2 also confirm the theoretical R-D behaviors.

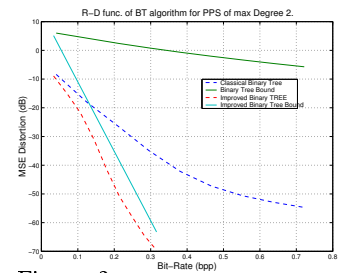


Figure 2: Theoretical and numerical R-D curves for the classical and improved binary tree algorithms.

IV. EXTENSION TO 2-D: QUAD TREE ALGORITHMS

Although the situation is much more open and complex in two dimensions (2-D), yet it is not hard to visualize the extension of the proposed 1-D coding scheme to the 2-D coding scheme. Since all the sub-algorithms developed for the 1-D scheme have a very nice equivalent in the 2-D world, e.g. the binary tree segmentation can be replaced by the quadtree segmentation and polynomial model can be replaced by the 2-D geometrical model consisting of two 2-D polynomials separated by a polynomial boundary. The Lagrangian optimization algorithm remains the same. The neighbor joint coding algorithm is a bit more involved but can be implemented efficiently. Therefore we can have an efficient quadtree based coding scheme for 2-D

signals with a reasonable computational complexity.

The classical quadtree algorithm: Our algorithm employs a quadtree segmentation followed by a coding algorithm on each image block in an operational R-D optimal sense. We employ an operational R-D optimization that is similar to the approach used in [8] in finding the best wavelet packet bases. The classical quadtree algorithm also exhibits the suboptimal performance due to the independent coding of the dependent nodes. For correcting the suboptimal behavior, we propose the improved quadtree algorithm.

The improved quadtree algorithm: The improved coding scheme employs the same previous classical quadtree coding scheme followed by the neighbor joint coding algorithm which, in fact, decides whether neighbors should be coded jointly or separately. The neighbor joint coding algorithm can be developed in a similar way as it was developed for the 1-D case.

R-D performance: It can be easily shown that for some simple image models, e.g. *Polygonal* model in which a polygonal boundary separates two polynomials, the improved quadtree coding scheme achieves the oracle performance ($D(R) \sim c_0 2^{-c_1 R}$) whereas the classical quadtree scheme performs suboptimally ($D(R) \sim 2^{-C\sqrt{R}}$) [3, 10].

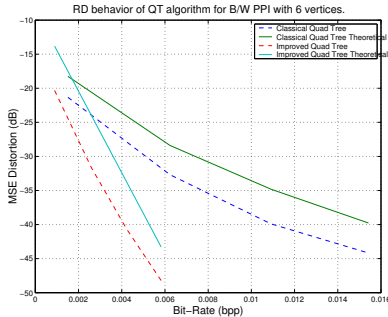
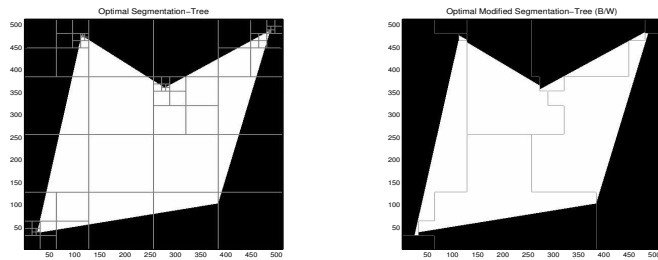


Figure 3: Theoretical (solid) and numerical (dotted) R-D curves for classical and improved quadtree algorithms for the B/W *Polygonal* image class.



(a) Classical quadtree segmentation.

(b) Improved quadtree segmentation.

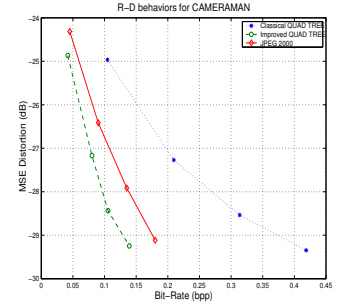
Figure 4: Examples of the quadtree representation.

V. DISCUSSION AND FURTHER WORK

Numerical results indicate that the proposed binary tree based coding scheme represents a nice way to model and approximate the piecewise polynomial signals or, more generally, piecewise smooth signals. Similar R-D results can also be achieved by the dynamic segmentation algorithm proposed in [6] with a large computational cost ($O(N^3)$) whereas our improved binary tree compression scheme achieves the desired R-D behavior with much lower computational cost ($O(N \log N)$). While dynamic programming methods do not work for 2-D problems, our binary tree based coding scheme can be easily extended to 2-D problems in the form of quadtree based coding scheme with a reasonable computational cost.



(a) Improved quad-tree (0.11 bpp).



(b) R-D performance.

Figure 5: Comparison of operational R-D curves using a wavelet coder (JPEG-2000) and quadtree coders for Cameraman image.

We have also derived the asymptotic R-D behavior of our tree based coding schemes for the piecewise polynomial functions. Numerical simulations (Figure 3) confirm that these algorithms achieve optimal performance if the input image fits the model exactly. In addition, preliminary simulations show that our quad-tree algorithm outperforms JPEG2000 also in the case of compression of real life images such as *Cameraman* (Figure 5). Our aim is to improve such a performance further by better compressing the texture present in images. Another issue is to understand whether the proposed quad-tree scheme provides an efficient and fast tool to detect and compress contours in object oriented multimedia applications. We would also like to explore the other application areas, e.g. image denoising and stereo image coding.

REFERENCES

- [1] A. Cohen, I. Daubechies, O.G. Guleryuz and M.T. Orchard, "On the importance of combining wavelet-based nonlinear approximation with coding strategies," *IEEE Trans. Inform. Th.*, July 2002.
- [2] D.L. Donoho, M. Vetterli, R.A. DeVore, and I. Daubechies, "Data compression and harmonic analysis," *IEEE Trans. Inform. Th.*, vol.44, no.6, pp. 2435-2476, October 1998.
- [3] M.N. Do, P.L. Dragotti, R. Shukla and M. Vetterli, "On the compression of two dimensional piecewise smooth functions," in *Proc. IEEE Int. Conf. on Image Proc. (ICIP)*, Thessaloniki, Greece, October 2001.
- [4] P.L. Dragotti and M. Vetterli, "Wavelet Transform footprints: catching singularities for compression and denoising," in *Proc. IEEE Int. Conf. on Image Proc.*, Vancouver, Canada, Sep. 2000.
- [5] S. Mallat, "A Wavelet Tour of Signal Processing," Academic Press, San Diego, CA, USA, 1997.
- [6] P. Prandoni and M. Vetterli, "Approximation and compression of piecewise smooth functions," *Phil. Trans. Royal Society London*, August 1999.
- [7] H. Radha, M. Vetterli, and R. Leonardi, "Image compression using binary space partitioning trees," *IEEE Trans. Image Proc.*, volume.5, no 12, pp. 1610 -1624, Dec. 1996.
- [8] K. Ramchandran and M. Vetterli, "Best wavelet packet bases in a rate distortion sense," *IEEE Trans. Image Proc.*, vol.2, no.2, pp. 160-175, April 1993.
- [9] Y. Shoham and A. Gersho, "Efficient bit allocation for an arbitrary set of quantizers," *IEEE Trans. Acoust. Speech Signal Processing* 36, 1445-1453, 1988.
- [10] R. Shukla, P.L. Dragotti, M.N. Do and M. Vetterli, "Improved quadtree algorithm based on joint coding for piecewise smooth image compression," accepted to the *IEEE International Conference on Multimedia*, Switzerland, August 2002.
- [11] X. Wu, "Image coding by adaptive tree structured segmentation," *IEEE Trans. Inform. Theory*, vol. 38, pp. 1755-1767, Nov. 1992.