WAVELET TRANSFORM FOOTPRINTS: CATCHING SINGULARITIES FOR COMPRESSION AND DENOISING

Pier Luigi Dragotti[1] *Martin Vetterli*[1,2]

[1] Laboratoire de Communications Audiovisuelles
Swiss Federal Institute of Technology, CH-1015 Lausanne, Switzerland.
[2] EECS Dept., UC Berkeley CA 947020
e-mail: {dragotti,vetterli}@lcavsun1.epfl.ch
`http://lcavwww.epfl.ch/~{dragotti,vetterli}/`

ABSTRACT

In recent years wavelets have been widely used for signal compression, image compression being a prime example, and for signal denoising. What makes wavelets such an attractive tool is their capability of representing both transient and stationary behaviors of a signal with few coefficients. In this paper we consider the problem of compressing and denoising a particular class of functions: piecewise polynomial signals. We show the limit of usual wavelet coders and present an alternative compression algorithm. The main innovation of the algorithm is that it tries to efficiently compress the significant coefficients of the wavelet decomposition rather then the zero coefficients as in usual coders.

The proposed algorithm can potentially be extended to more general signals and represents an effective solution to problems like signal denoising and image compression.

1. INTRODUCTION

Wavelets are known to be efficient in representing piecewise smooth functions. Away from singularities, the inner product between a wavelet (with a number of zero moments) and a smooth function will be either zero or very small [8, 9]. At singular points, a finite number of wavelets concentrated around the discontinuity lead to non-zero inner products. This is in contrast with Fourier series where discontinuities lead to many larger coefficients. This ability of wavelets expansions to capture both smooth and singular parts of a signal has been used in many applications, including denoising and compression.

But as far as compression is concerned, in [5] it was shown that for the class of piecewise polynomial signals wavelets are far from reaching the asymptotic rate-distortion behavior, since a wavelet coder behaves as :

$$D_w(R) = \hat{C}_w(1 + \alpha\sqrt{C_w R})2^{-\sqrt{C_w R}},$$

while, if we suppose that the degrees of polynomials and the number of singularities are provided by an oracle [5] then the rate distortion curve is:

$$D_p(R) = \hat{C}_p 2^{-C_p R}.$$

Pier Luigi Dragotti was supported by Fonds National Suisse project number 21-52439.97

It is the purpose of this paper to improve the wavelet coder for piecewise polynomial signals so as to reduce the gap with the oracle method. We start by noting that the poor behavior of the usual wavelet coder comes from the independent coding of wavelet coefficients around discontinuities. However, these coefficients are highly dependent across scales. We thus introduce the notion of "footprints", which are the traces left by singularities in the wavelet domain. By coding these footprints instead of the individual wavelet coefficients, we are able to improve state of the art of wavelet coders.

It is of interest to compare footprints with zerotrees [7]: zerotrees indicate absence of singularities, while footprints describe singularities. Depending on the density of singularities, footprints can be more efficient than zerotrees. But the real difference appears around singularities, which are not well represented by zerotrees (instead, a number of non-zero coefficients are coded independently) while they can be more efficiently represented by footprints.

Since wavelets are often used for compression of images, the notion of footprints can be used there as well. It is to be noted that singularities in images are typically one dimensional (a contour), and thus wavelets are not necessarily the best bases for representing such singularities.

2. FROM DISCONTINUITY CHARACTERIZATION TO FOOTPRINTS

We say that a function f is piecewise smooth if it is a regular function (i.e. n times continuously differentiable function) everywhere, except for a finite number of points of discontinuity (breakpoints). Real life signals are often considered piecewise smooth and usually the essential information of the signal is carried by the discontinuities. Thus their efficient characterization is the key problem in signal compression or denoising.

Fourier transform is not a good tool to represent piecewise smooth functions, since the spectrum decay of the Fourier transform depends on the global regularity of the signal and thus it is conditioned by the discontinuities.

On the other hand wavelets are able to characterize the local regularity of a function. In particular it can be shown that if we take a wavelet with enough vanishing moments then the speed of decay of the wavelet coefficients across scales depends on the local Lipschitz regularity of the signal [4]. So around the discontinuities the wavelet coefficients have a slower decay (the Lipschitz coefficient is small) while around the regular part of the signal their decay is faster (rather large Lipschitz coefficient). In particular if the smooth signal is a polynomial of maximum degree n and if the wavelet has at least $n+1$ vanishing moments then the inner product between the wavelet and the polynomial will be exactly zero. So for piecewise polynomial signals the wavelet coefficients are non zero only around discontinuities. This is equivalent to saying that all the information about the signal is contained in the few significant wavelet coefficients around the discontinuities.

Similar properties apply to discrete signals and discrete wavelet bases. Here again the discrete wavelet coefficients away from discontinuities are very small or zero. In Figure 1 we show the wavelet decomposition (using critically subsampled filter banks) of a discrete time piecewise polynomial signal. We can clearly see that the significant coefficients are only around discontinuities.

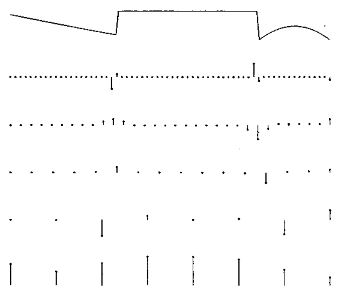

Fig. 1. Wavelet decomposition of a piecewise polynomial function.The first 4 level are wavelet coefficients, while the last level are scaling coefficients

If we use a non redundant discrete wavelet basis to decompose our signal then the wavelet coefficients are not shift invariant. This means that the same discontinuity at two different positions will be represented by different wavelet coefficients.

In case of piecewise polynomial functions and non redundant wavelet decomposition the significant coefficients have some peculiar property:

- The significant coefficients are spatially located around the signal discontinuities.
- At each discontinuity there are no more than $L-1$ significant coefficients per scale, where L is the length of the wavelet filter.
- The significant coefficients generated by a discontinuity are highly correlated (Actually, they are deterministic, since they only depend on the characteristic of the discontinuity).

In traditional wavelet based compression and denoising algorithm these coefficients are processed independently . In case of compression they are scalar quantized [1], while in case of denoising they are thresholded independently. However they should be gathered in a vector and jointly processed. For this reason we introduce the notion of *footprint* which is a vector containing all significant wavelet coefficients across scales around a discontinuity. For instance if our wavelet filter has length L and we have J wavelet decomposition levels then the *footprint* of a discontinuity is a vector of dimension $(J+1)\times(L-1)$ containing $L-1$ wavelet coefficients at each scale in the position corresponding to the discontinuity position plus the $L-1$ scaling coefficients at the same position.

Moreover:

1. Footprints are not shift invariant.
2. Each footprint implicitly contains all the information related to the discontinuity that generated it: amplitude of the discontinuity and characteristic of the two polynomials that are around the discontinuity.

The notion of footprint is valid also for piecewise smooth functions, however in this case the knowledge of all the signal footprints is not enough to completely describe the signal, since the wavelet coefficients away from discontinuities are small but not zero.

Finally the definition of footprint can be naturally extended to the case of redundant wavelet decomposition (i.e., oversampled filter banks). In this case footprints become shift invariant.

3. A WAVELET BASED CODER FOR PIECEWISE POLYNOMIAL FUNCTIONS

The compression algorithm we propose takes the results of the previous section as a starting point. Since the knowledge of footprints is enough to completely describe our signals, our compression algorithm consists in first finding the footprints of the signal and then coding them efficiently.

Starting from the finest scale of the wavelet decomposition where discontinuities can be detected, the coder looks for significant coefficients, when one is found its position is scalar quantized, the corresponding footprint is generated and then it is vector quantized.

Notice that given the positions of the discontinuities and the corresponding footprints which describe them, the signal is completely determined except for its mean. Consider, for example, the two discontinuities represented in Fig.1. With the first footprint we define the characteristic of the first and the second polynomial and of the amplitude of the first discontinuity, with the second footprint we define the amplitude of the second discontinuity and the characteristic of the third polynomial. Thus the only information we need to completely define the signal is its mean.

In terms of compression given a total bit budget R_{tot} and a signal characterized by no more than M discontinuities, we allocate R_p bits to code each position of a discontinuity (uniquely defined by the position of the significant coefficients in the finest level), R_f bits to vector quantize each footprint and R_m bits to code the mean. Based on the results in [5], the allocation strategy is approximately defined by the following equations:

$$R_p = \frac{2}{2M(N+2)+1}R_{tot} - 2MK + \frac{2}{2M(N+2)+1}$$

$$R_f = \frac{2(N+1)}{2M(N+2)+1}R_{tot} + (2M+1)K + \frac{2(N+1)}{2M(N+2)+1} \qquad (1)$$

$$R_m = \frac{1}{2M(N+2)+1}R_{tot} - MK - \frac{2M(N+2)}{2M(N+2)+1}$$

with $K=(N+1)/(2M(N+2)+1)\log_2(N+1)$ and N representing the maximum degree of any polynomial in the signal.

The vector quantizer is constructed using the generalized Lloyd-Max algorithm [3]. For each of the 2^{R_p} possible quantized positions the vector quantizer constructs a different codebook which contains the 2^{R_f} footprints related to the best breakpoints in the

[1] Actually, in some compression algorithms the significant coefficients are compressed using context based coders, here the correlation between significant coefficients is partially exploited

mean square error sense for that position. It can be shown that the codewords generated in this way keep belonging to the space of piecewise polynomial functions, so they keep representing discontinuities between two polynomials.

Notice that since footprints are not shift invariant, it is not possible to compare footprints related to different positions; in fact the same discontinuity but in two different locations generates two different footprints, so the square error between them is not zero. Now if the signal we have to code has a discontinuity at a position N which is one of the 2^{R_p} positions for which we have designed a codebook then we can use the corresponding codebook to find the best footprint for that discontinuity. On the other hand if the position N is not one of these 2^{R_p} positions than it is not possible to use any of the designed codebooks, since the distances between footprints in the codebook and the footprint we have to code are misleading. To overcome this problem a shifted version of the codebook related to the quantized position closest to N is considered for vector quantization. Basically if the position N is going to be scalar quantized to the position $\hat{N}$ then the corresponding footprint will be vector quantized with a shifted version of the footprint codebook related to the position $\hat{N}$.

The following table presents a pseudo-code description of our coder.

Off line:

1. Given the statistical characteristic of the input signals, chose the allocation strategy defined by equation 1

2.Choose the number J of wavelet decomposition levels.

3. For each of the 2^{R_p} positions construct a different codebook using the generalized Lloyd-Max algorithm.

On line:

Encoding:

1. Scalar quantize the mean of the signal to be compressed and send it to the decoder.

2. Make a J level wavelet decomposition of the zero mean signal.

3. Using a discontinuity detector in wavelet domain (e.g. looking at the finest scale), find the discontinuity position.

4. Find the best footprint for that discontinuity.

5. Send the position and the footprint indices to the decoder.

6. Repeat step 3-4-5 until all discontinuities are coded.

The decoder operates directly in the time domain. With the first position/footprint indices received it estimates the position and the characteristics of the first discontinuity including the characteristics of the input and output polynomials. For each new pair position/footprint of indices it estimates the amplitude of the discontinuity and only the characteristics of the output polynomial and concatenate it to the old decoded discontinuities. When all the discontinuities are coded the quantized mean is added to the reconstructed signal.

It is interesting to notice that our proposed compression approach is quite different from traditional wavelet based compression algorithms. In fact in traditional approaches the main interest is in representing the long sequences of zeros or small wavelet coefficients efficiently (zerotrees being a main example [7]), while our coder is only interested in the representation of the significant coefficients.

As it will be shown in the next section, the proposed algorithm is very efficient not only for signal compression but also for denoising.

4. EXPERIMENTAL RESULTS

In this section we want to assess the performance of our algorithm. We consider piecewise polynomial signal of maximum degree $N = 1$ (piecewise linear), the discontinuity positions are uniformly distributed on the time interval. We will show that our algorithm outperforms a traditional compression algorithm like SPITH [6].

Moreover the concept of footprint can be used in other applications like signal denoising and also here the results are encouraging.

As a first experiment we compare the proposed algorithm with one of the most competitive wavelet based compression algorithm: SPITH ([6]). SPITH has been designed to work on images, so to make our comparison we have modified it to work on 1-dimensional signals. In table 1 we compare the PSNR at different bit rates for the case of piecewise linear signals and no more than $M = 3$ discontinuities. Figures 2 and 3 show an example of reconstructed signal using SPITH or our algorithm. The algorithm based on footprints gives better results either in terms of PSNR or visually.

	0.07b/p	0.09b/p	0.1b/p
SPITH	13.5dB	18.6dB	20.1dB
Our coder	23.1dB	24.1dB	25.6dB

Table 1. PSNR results.

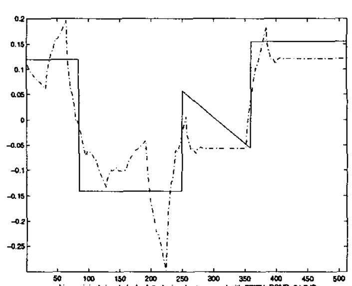

Fig. 2. Line: original signal. Dash-dotted: signal compressed using SPITH algorithm. PSNR=24.7dB, rate=0.15bps

This comparison is not completely fair since SPITH has not been designed to compress piecewise polynomial signals, anyway

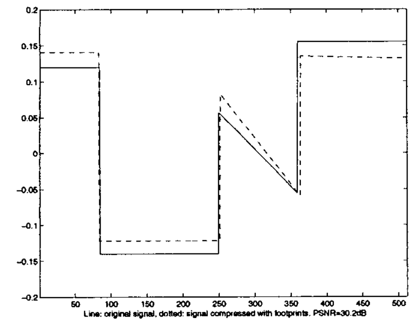

Fig. 3. Line: original signal. Dashed: signal compressed using footprints based algorithm. PSNR=30.2dB, rate=0.1bps

we think it is useful to show that our algorithm can outperform traditional wavelet based compression algorithms.

Footprints can also be extremely effective for denoising. In a first approach the noisy piecewise constant signal is transformed into the wavelet domain. The wavelet coefficients are hard thresholded and then the signal is denoised and compressed using the algorithm presented in the previous section. In Figure 4 we show the results and compare the proposed approach with the usual denoising algorithm based on soft-thresholding [1]. On average our

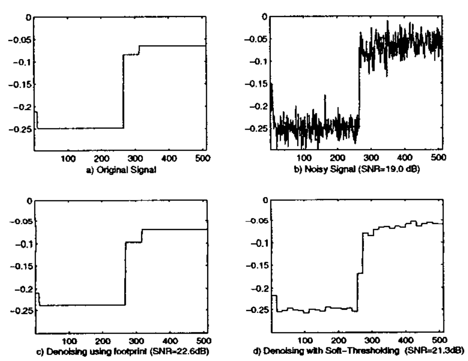

Fig. 4. a) Original Signal, b) Noisy Signal (SNR=19.0), c) Denoised Signal using footprints (SNR=22.6dB), d) Denoised Signal with Soft-Thresholding (SNR=21.3dB)

algorithm outperforms the Soft-Thresholding algorithm by about $3dB$. However footprint denoising can be done without compression. Further results on this topic can be found in [2].

Finally we would like to know if these results can be extended to real-world signals like images. At the moment it is not clear how to extend the notion of footprint to the two-dimensional non-separable case. However if we decide to compress each line of an image independently then the results are quite encouraging. In Figure 5 we show a single line of the image "Lena" and its piecewise constant approximation obtained with our algorithm. We believe that for increasing bit rates the algorithm will better represent the polynomial behavior underlying the image, while the finer details could be represented as an additive noise-like residual. However it is too early to say if this kind of approach can lead to a competitive image compression algorithm. A true two dimensional version of

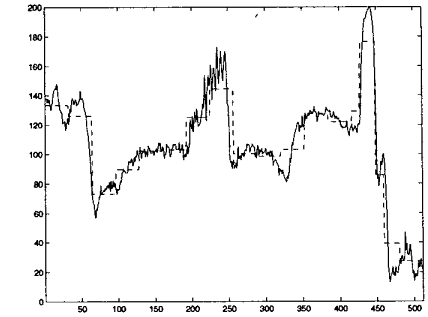

Fig. 5. Piecewise constant approximation of a line of Lena

footprints is a topic under investigation.

5. CONCLUSIONS

In this work we have addressed the problem of efficiently compressing piecewise polynomial signals. We have analyzed the correlation existing between wavelet coefficients and introduced the notion of footprints. Based on the idea of footprints, we have designed a new wavelet coder that outperforms traditional wavelet based compression algorithms.

Moreover the concept of footprint can be used also in other context like signal denoising. First results are promising. We believe that footprints can be successfully used also for compression of piecewise smooth signals and two-dimensional signals like images. These topics are under investigation.

6. REFERENCES

[1] D.L.Donoho "De-Noising by soft thresholding," IEEE Trans. Info. Theory 41(3): 613-627, May 1995.

[2] P.L. Dragotti and M. Vetterli "Shift-Invariant Gibbs Free Denoising Algorithm based on Wavelet Transform Footprints", Accepted for SPIE2000, Wavelet Application in Signal and Image Processing, August 2000.

[3] A. Gersho and R. Gray *Vector Quantization and signal compression,* Kluwer Academic Publishers, Boston, 1992.

[4] S.Mallat *A wavelet tour of signal processing,* Academic Press, 1998.

[5] P. Prandoni and M. Vetterli "Approximation and Compression of Piecewise Smooth Functions," Phil. Trans. R.Soc.Lond. 1999.

[6] A.Said and W.A.Pearlman "A New, Fast, and Efficient Image Codec Based on Set Partitioning in Hierarchical Trees", IEEE Trans. Circuits and System for Video Technology, Vol.6, No.3, pp. 243-249, Jun. 1996.

[7] J.M. Shapiro, "Embedded image coding using zerotrees of wavelets coefficients," IEEE Trans. Signal Processing, vol. 41 pp.3445-3462, Dec. 1993.

[8] G. Strang and T. Nguyen *Wavelets and Filter Banks* Wellesley-Cambridge Press.

[9] M.Vetterli and J. Kovačević, *Wavelets and Subband Coding,* Prentice Hall, 1995.