A Method of Denoising and Reconstructing Audio Signals

Jonathan Berger
The Center for Studies in Music Technology
Yale University
jberger@alice.music.yale.edu

Ronald R. Coifman
Department of Mathematics
Yale University
coifman@jules.math.yale.edu

Maxim J. Goldberg
Physical Sciences Department
York College of PA
mgoldberg@yorkcol.edu

June 12, 1994

Abstract

In this paper we present a description of the application of a denoising algorithm for removing noise from music. We describe the library of orthonormal bases used, and briefly describe the method of choosing the optimal basis for the signal and the process of separating the signal into coherent and noisy parts. We then describe several methods of breaking a large music signal into windows, and of combining these windows in order to diminish the audible effects of the windows' edges. We conclude by considering denoising approaches used within a window.

Introduction

We primarily considered two recordings: one is a copy of a wax cylinder recording of Johannes Brahms performing his Hungarian Dance, no. 1, in g minor, the other is a portion of a piano vocal arrangement of an aria from Puccini's Tosca, sung by Enrico Caruso in 1903 and available on a Pearl CD.

The paper is organized as follows. We first present an overview of the denoising algorithm, describing the library of orthonormal bases used, the method of choosing the optimal basis for the signal, and the process of separating the signal into coherent and noisy parts. We then describe several methods of breaking a large music signal into windows, and of then combining these windows to decrease edge effects. We finally discuss different denoising approaches used within a window, including the consideration of subjective perception criteria.

A detailed description of this work is given in [Berger, Coifman, Goldberg]. Our algorithm is closely based on the work of R. Coifman and V. Wickerhauser on entropy–based algorithms, see [Coifman, Wickerhauser, 92], and on the work of R. Coifman and F. Majid on the denoising process, see [Coifman, Majid, 93].

The denoising experiments described have been run using software developed during 1992–1993, which is based on the Adapted Waveform Analysis Library (awa), [Wickerhauser, 91–92] .

This work was begun in 1992 at Yale University.

The Denoising Algorithm

A general description of the denoising algorithm is as follows:

One starts with a library of orthonormal bases (we will give some examples of such libraries later). A signal is expanded in each basis, and a cost assigned to the expansion. A useful cost function, the Shannon entropy, measures the efficiency of the representation of the signal in a particular basis. The basis giving rise to the least cost is chosen, the coefficients are ordered by magnitude, and a number of the leading terms is kept as the coherent part, based on a predetermined threshold cost of the remaining terms. These residual terms by definition constitute the noisy part of the signal, and can be treated as a new signal which can in turn be expanded and separated into its coherent and noisy components. Thus, at each iteration, a coherent part is extracted from the signal currently considered, and the leftover noise is then treated in turn. The coherent parts, one from each iteration, can be added together to produce the total coherent portion.

Let us now give some more details of the above process. We describe the orthonormal bases that were included in our library; due to space restrictions, however, we do not elaborate on the cost function nor on the method of separating the signal into coherent and noisy parts, beyond what has already been mentioned above.

One class of bases comes from the local trigonometric bases of Coifman and Meyer (see [Auscher, et al, 92] for a complete description of such bases). A straightforward local trigonometric procedure would be done as follows. The signal is split in the middle, and the left and right halves are expanded in some discrete trigonometric basis, discrete sine or cosine. For a description of discrete cosine transforms see [Rao, Yip, 90]. We can then split the left half into two equal parts, and do a local trigonometric transform on these two pieces. Similarly, the right half is divided into two equal segments. Thus we construct a binary tree of coefficients where the root is the original signal, the left child is the left half of the signal, the right child is the right half of the signal, and so on. To each node we associate a local cosine or local sine expansion of that portion of the signal. The same transform, once chosen, is used for each node. We continue $\log_2$ (length of signal) times, padding the signal with zeros if necessary to assure that the signal length is a power of 2. In practice, a long signal is split into several windows of a manageable size, and the last window is padded with zeros if necessary.

The procedure described suffers from a serious drawback when reconstruction is attempted from a subset of coefficients (from the chosen basis). Namely, the abrupt splitting creates sharp cuts, and these give rise to lingering artifacts when the trigonometric transform is performed.

One way to avoid this problem is to multiply the signal by smoothly decaying bells which are supported on roughly the same entries as the corresponding characteristic functions. However, this is impossible to do without overlap, and thus an orthonormal basis for the whole signal cannot be built in a straightforward way from the bases for the left and right descendants. A discovery by Coifman and Meyer, see [Auscher, et al, 92], allows one to retain the features of successive segmentation and to retain orthonormality. The projection from the whole segment into the left and right halves is not just multiplication by a function, but a multiplication by a smooth bell combined with a process of making the resulting subsignal even on one side and odd on the other. This is a process called folding. Such folding is done only near the ends of the subsegment and the width affected by the folding can be made as small as desired in relation to the width of the subsegment. See [Auscher, et al, 92] for more details.

In our experiments, we have constructed other classes of orthonormal bases, also arising from binary trees. One class is obtained by successive application of a Quadrature Mirror Filter (QMF). Such a class includes the wavelet basis (corresponsing to the chosen QMF) as a particular case (see [Daubechies, 88] for background on wavelets and QMF's; the construction is described in [Coifman, Wickerhauser, in preparation]). Another class comes from applying a discrete trigonometric transform (we used discrete sine) to (a window of) a signal, and then building a binary tree of the transformed signal using either a local trignometric basis or a QMF.

Segmentation and Averaging

Segmentation of the signal using straightforward cuts creates sharp discontinuities, resulting in periodic clicks on the time side and whistles on the frequency side. One promising approach to alleviate this problem has been to attempt to mimic what is already being done internally by the local trigonometric tree construction. Namely, to multiply the signal by smooth, compactly supported bells, and then fold adjacent edges of two adjacent windows (one side in an even, the other in an odd way). Unfortunately, the folding introduces problems at the window boundaries.

Although undoubtedly some noise cancelation occurs, some of the noise is reinforced when added together from the edges which are folded into each other. Thus spikes are created at the window junctures which are, alas, very audible to the ear.

An alternative method involves considering an expanded window rather than performing the denoising algorithm on each individual window. This is the process we chose to incorporate into our software. In this procedure, we look at a signal centered at the window we wish to denoise but protruding beyond. Let us suppose, for example, that we consider a signal 4 times the width of the window. As a first step, we multiply the original signal by a characteristic function of support 4 times the width of the window, centered at the window. The resulting expanded window is then multiplied by a smooth bell, supported on the expanded window, and equal to 1 on the original window, and even slightly beyond. This attenuated, expanded window is the one then denoised. From the resulting coherent portion, only a central

part is picked, one that is even shorter than the original window. This core part should suffer relatively little from edge effects: the bell is smooth and the attenuation is done far away from the core part, so even if edge artifacts are left, they should be insignificant at the core. The cores, as constructed above, leave gaps in the signal. Thus, the entire signal is then shifted one half window width, and the same procedure is carried out on it. The cores from the first shift are smoothly merged with cores from the second shift, using a bell as a gluing function. This smooth merger allows the coherent structure in one core to merge gradually with the coherent structure in the next core, so that the coherent signal's character does not change abruptly.

To reduce the effect of the dyadic grids imposed on the signal by the binary construction, we tried processing the signal shifted several times by padding it with varying numbers of leading zeros. By so doing the window cuts and internal subdivisions fall at different places. We then shift back the coherent signal extracted from each shifted signal. Whatever numerical errors arise in the reconstruction due to these partitions are shifted to different temporal locations. The underlying true signal which we are trying to extract is not changed by these shifts. We then average the various coherent signals obtained, thus dividing the introduced errors by the number of shifts. Since the errors due to the grids will occur in diffent places, they do not reinforce one another.

In our experiments with music, we eventually settled on using only one tree, rather than several, in our library, and extracted the best basis from it. This was the local sine tree obtained from the discrete sine transform of a(n expanded) window. Local sine on the spectrum is well suited to zero in on frequencies with much activity, where high resolution is important, and gives the most aesthetically acceptable result.

We now describe the procedure of frequency averaging. For music and speech, important frequencies tend not to change too much from window to window. The forced dyadic nature of the local sine basis will therefore tend to introduce similar artifacts in window after window. Upon retransforming to the time side, these artifacts become persistent, and annoying, whistles. By examining the spectrum of sample windows, we noticed that the last 200 or 300 entries (from an expanded window of size 4096), the high frequencies, were in size much less than the lower frequencies. To do frequency shifts, we take the vector of frequencies, shift it to the right, losing shift number of rightmost entries, and pad the front with shift number of 0's. We then denoise using local sine, and shift back, padding now the right end with 0's. We do several such shifts and take their average, and finally go back to the time side by the inverse sine transform. Such averaging seems to be even more effective than averaging in time for eliminating annoying sounds. In practice it seems useful to perform several shifts of both types.

Denoising One Window

One simple improvement to the denoising procedure is to use a variation on coefficient shrinkage, proposed in [Donoho, Johnstone, 92]. Suppose in the denoising procedure, the first k coefficients, $y_1, ...y_k$, ordered by magnitude, in the expansion of a window or its trigonometric transform, are to be selected for the coherent signal. Rather than choosing them, we find an index j so that y_j is, say, roughly twice the size of y_k. Coefficients $y_1, ...y_{j-1}$ are not changed in any way. Coefficients y_j to y_k are smoothly decreased, in absolute value, by a bell times $|y_{k+1}|$. Thus, a noisy part is subtracted from the smaller coherent coefficients, where it has a comparatively large effect, while the larger coefficients are kept in their entirety for the coherent component. The subtraction is done in a such a way as to smoothly decrease the coefficients to zero. Soft thresholding produces audible improvements when used in processing Caruso's singing.

We have also implemented an entirely different process in which data on the frequency dependence of amplitude perception is incorporated into the method. Extensive studies have been conducted on subjective perception of loundess as a function of pitch and sound pressure, see [Boff, et al, 86]. Experimental equal–loudness contours have been published, as well as empirical power laws relating loudness, frequency, and sound pressure, see [Boff, et al, 86], Chapter 15. We used the formula

$$L = k_f P^{.6},$$

where L is loundess in sones, P is pressure in micorpascals, and k_f is a coefficient that depends on frequency.

The sone scale is useful since it is linear, i.e. if a sound has twice the sone level of another sound, it is perceived to be twice as loud. Using the equal–loudness contours, it is possible to estimate k_f for each frequency.

One promising approach tried works as follows: we take the vector of frequencies and find the best basis expansion using local sine in the usual way. Then, before ordering the coefficients, we weight them to reflect the perceptual data. So a high frequency, small

amplitude component may be listed before a low frequency component with a larger amplitude after this weighted ordering. Then we unscale. The coefficients are now ordered only in the weighted sense, but their order, not they themselves, has been changed. Now we denoise, stopping when the entropy of the tail just exceeds the threshold. Note that here it does not make sense to apply soft thresholding. In windows in which singing occurs, the reconstruction seems to be equally good to that obtained using ordinary ordering, but a higher compression is achieved. In windows in which noise is the predominant feature, the denoising procedure seems to break down and pick nearly the entire signal as being coherent including within all the noise. Thus there are periods of clear singing with unnerving spurts of noise in between. It may be that the treshold criterion is too primitive, and something more delicate and more adapted to each particular window should be used.

Summary

The denoising algorithm has proved to be a very useful technique for removing noise from musical signals. The use of the entropy function to select an optimal local trigonometric basis on the signal's spectrum creates a basis well suited to the particular signal and concentrates attention on its important frequencies. Selecting the coherent portion is then done using coefficients whose size reflects their importance in the signal's coherent structure. We have devised and experimented with several techniques to make the output as musically clean as possible, without removing musical content. Some problems remain, such as isolated clicks and whistles, and uneven representation given to some frequencies. Denoising using criteria suited to the application, such as subjective auditory perception for processing music, seems a promising possiblity. Another possibility is denoising frequency bands separately, rather than the entire spectrum of a window.

References

[Auscher, et al, 92] P. Auscher, G. Weiss, and M.V. Wickerhauser, *Local Sine and Cosine Bases of Coifman and Meyer and the Construction of Smooth Wavelets*, Wavelets: A Tutorial in Theory and Applications (C. Chui ed.), Academic Press, Inc., 1992.

[Berger, Coifman, Goldberg] J. Berger, R. Coifman, M. Goldberg, *Removing Noise from Music using Local Trigonometric Bases and Wavelet Packets*, Journal of the Audio Engineering Society, forthcoming.

[Berger, Coifman, Goldberg, Nichols] J. Berger, R. Coifman, M. Goldberg, C. Nichols, *Incorporating Psychoacoustic Studies in a Denoising Algorithm*, in preparation.

[Boff, et al, 86] K. Boff, L. Kaufman, J. Thomas, eds., Handbook of Perception and Human Performance, Vol. 1: Sensory Processes and Perception, John Wiley and Sons, 1986.

[Coifman, Majid, 93] R. Coifman, F. Majid, *Adapted Waveform Analysis and Denoising*, in Progress in Wavelet Analysis and Applications, edited by Y. Meyer and S. Roques, Proceedings of the International Conference "Wavelets and Applications", Toulouse, France, Editions Frontieres, pp. 63–76, 1993.

[Coifman, Meyer, et al, 93] R. Coifman, Y. Meyer, S. Quake, M.V. Wickerhuaser, *Signal Processing and Compression with Wavelet Packets*, in Progress in Wavelet Analysis and Applications, edited by Y. Meyer and S. Roques, Proceedings of the International Conference "Wavelets and Applications", Toulouse, France, Editions Frontieres, pp. 77–93, 1993.

[Coifman, Wickerhauser, 92] R.R. Coifman and M.V. Wickerhauser, *Entropy-based Algorithms for Best Basis Selection*, IEEE Trans. Inform. Theory, 32:712–718, March 1992.

[Coifman, Wickerhauser, in preparation] R.R. Coifman and M.V. Wickerhauser, *Wavelets and Adapted Wave form Analysis; A Toolkit for Signal Processing and Numerical Analysis*, in preparation.

[Daubechies, 88] I. Daubechies, *Orthonormal Bases of Compactly Supported Wavelets*, Comm. on Pure and Applied Math., Vol XLI, 909–996, 1988.

[Donoho, Johnstone, 92] D.Donoho and I. Johnstone, *Ideal Spatial Adaptation via Wavelet Shrinkage*, Dept. of Statistics, Stanford University, preprint, 1992.

[Rao, Yip, 90] K. Rao, P. Yip, Discrete Cosine Transform: Algorithms, Advantages, Applications, Academic Press, Inc., 1990.

[Wickerhauser, 91–92]
V.M. Wickerhauser, Adapted Waveform Analysis Library (software package for wavelet packet and local trigonometric decompositions), Wickerhauser Consulting, 1991–1992.