# OPTIMAL TIME SEGMENTATION FOR SIGNAL MODELING AND COMPRESSION

*Paolo Prandoni*[1] *Michael Goodwin*[2] *Martin Vetterli*[1,2]

[1] LCAV, Ecole Polytechnique Fédérale de Lausanne, Switzerland
[2] EECS, University of California, Berkeley, USA
email: prandoni@de.epfl.ch, michaelg@eecs.berkeley.edu, vetterli@de.epfl.ch

## ABSTRACT

The idea of optimal joint time segmentation and resource allocation for signal modeling is explored with respect to arbitrary segmentations and arbitrary representation schemes. When the chosen signal modeling techniques can be quantified in terms of a cost function which is additive over distinct segments, a dynamic programming approach guarantees the global optimality of the scheme while keeping the computational requirements of the algorithm sufficiently low. Two immediate applications of the algorithm to LPC speech coding and to sinusoidal modeling of musical signals are presented.

## 1. INTRODUCTION

The topic of adaptive best bases [1, 2] is concerned with finding the best linear transform, or set of linear transforms, for a given signal. Various algorithms have been proposed, and have been applied to time-frequency analysis and signal compression. The algorithms are based on tree pruning for dyadic time segmentation [3] or on dynamic programming for arbitrary time segmentation [4].

In this paper we explore further the possibilities of adaptive representations by using more general models, in particular linear prediction and sinusoidal modeling. The main goal is to achieve better segmentations of time, and better local models. The search is done in an operational rate-distortion (R/D) framework. That is, we find the best possible joint segmentation and coding given the set of possible partitions and coding algorithms. A first application is related to the LPC coding of speech: the algorithm manages to redistribute the coding resources to yield an overall data rate reduction at basically no cost in terms of speech quality. As a further application, we also consider sinusoidal modeling of musical signals with time-varying windows, thus solving the pre-echo problem.

## 2. PROBLEM STATEMENT

In compression problems, it is almost always the case that the data to be processed possesses non-stationary characteristics. In general, the problem of time-varying data is addressed by means of a uniform "compromise" tiling of the time axis: an analysis window is designed as a tradeoff between achieved locality and efficiency of representation, and the data is segmented accordingly. The window design problem is usually ad-hoc in the sense that it relies on *a priori* information on the physical properties of the signal. In LPC speech coding, for instance, a frame size of 16-20 ms is usually recommended [5].

Once the input data is thus suitably segmented, the bit-allocation problem can be dealt with in several ways; global optimality cannot however be attained if the segmentation process is not part of the R/D optimization process.

In general terms, the global optimization problem can be stated as such: let $X$ be the data, $S$ the set of all possible segmentations applicable to the data, and $P$ the set of all allowed coding templates for all possible data segments; here, a coding template could be a particular coding scheme (LPC, transform coding, sinusoidal modeling, VQ), a particular quantizer choice, or any other choice of data representation, yielding an arbitrary R/D curve. Each $\sigma \in S$ splits the data into $N_\sigma$ segments; let $P_\sigma$ be the set of all possible choices of coding templates for the $N_\sigma$ data segments; clearly, $P_\sigma \subseteq P^{N_\sigma}$.

Given a total allowed rate of $R_{\max}$, we seek to solve

$$\min_{\sigma \in S} \min_{p \in P_\sigma} \{D(\sigma, p, X)\}$$

subject to

$$R(\sigma, p, X) \leq R_{\max},$$

where $D$ and $R$ are the overall distortion and rate obtained by appling $\sigma$ and $p$ to $X$. If the segments defined by $\sigma$ are disjoint, and rate and distortion are additive over disjoint segments, it is convenient to formulate the above constrained problem in its equivalent unconstrained form. Define the minimum Lagrangian cost of coding the $i$-th segment in partion $\sigma$ as

$$J(\lambda, X_i) = \min_{q \in P} \{D(q, X_i) + \lambda R(q, X_i)\}, \qquad (1)$$

where $D(q, X_i)$ is the distortion obtained by coding segment $X_i$ in partition $\sigma$ with coder $q$, and $R(q, X_i)$ is the corresponding rate. It follows that the equivalent unconstrained problem can be stated as finding the minimum total Lagrangian cost

$$\hat{J}_S(\lambda^*) = \min_{\sigma \in S} \{\sum_{i=1}^{N_\sigma} J(\lambda^*, X_i)\}, \qquad (2)$$

where $\lambda^*$ is the (initially unknown) optimal operating slope on the R/D curve which can be found by iteration. If $S$ possesses some form of hierarchical structure, the previous minimization can be carried out very efficiently by means of dynamic programming.

## 3. INCREMENTAL ALGORITHM FOR DYNAMIC SEGMENTATION

Following the lines exposed in [4], we will now briefly review how to efficiently explore the space of possible segmentations/resource allocation. For a data segment $X$, we define a minimal analysis block of length $L$ samples, which we will call a *cell*, and we assume that $X$ has $M$ cells, or $ML$ samples. Let $S$ be the set of all possible partitions of $X$ so that each segment is composed of an arbitrary number of cells. There are $2^{(M-1)}$ such partitions, ranging from

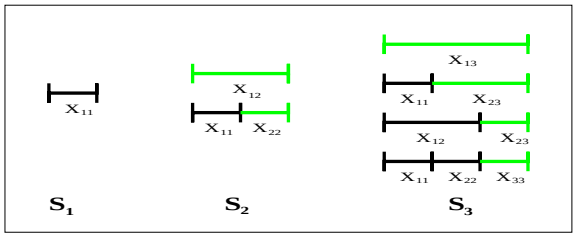


Figure 1. Exploring the segmentation space: incremental construction of $S_M, M = 3$.

taking $X$ as a whole to splitting $X$ into $M$ segments, each one cell long. Call $X_{jk}$ the segment from cell $j$ to cell $k$. The set $S$ can be built incrementally: let $S_0 = \emptyset$; at each step $n$, $1 \leq n \leq M$, form $S_n$ by "extending" all the partitions in $S_k$ by segment $X_{kn}$, for all $k$ from 0 to $n-1$ (see Figure 1, where the extending segments are drawn in gray); in the end $S = S_M$. The dynamic programming algorithm explores the space defined by $S$ exploiting the fact that, for each element of $S$, the total Lagrangian cost is the sum of the cost of coding the "extension" $X_{kM}$ plus the cost of coding an element of the subpartition defined by $S_k$, for some $k$; by the optimality principle, the minimum cost among all the elements of $S$ which share the same "extension" is found by taking the minimum over the space defined by $S_k$. Consequently, we can rewrite (2) as

$$\hat{J}_S(\lambda) = \min_{0 \leq k \leq M} \{\hat{J}_{S_k}(\lambda) + J(\lambda, X_{kM})\}. \quad (3)$$

This defines an incremental algorithm which, for a given operating point $\lambda$, yields the optimal segmentation and the minimum total Lagrangian cost in a number of steps which is only quadratic in $M$; also, the algorithm is only linear in $M$ in term of storage reqirements. We start by computing $\hat{J}_0(\lambda) = 0$, $\hat{J}_1(\lambda) = J(\lambda, X_{11})$ and then, at each step $n$, we compute

$$\hat{J}_n(\lambda) = \min_{0 \leq j \leq n} \{\hat{J}_k(\lambda) + J(\lambda, X_{kn})\}$$

until $n = M$. At each step we need only keep track of the newly computed values of $\hat{J}_k(\lambda)$ for $k$ from 0 to $n$ and of the value of $j$ yielding the minimum.

## 4. APPLICATIONS

### 4.1. LPC Speech Coding

Linear predictors require stationarity of the input signal; in speech coding the data is segmented using a fixed-size analysis window spanning 16-20 ms of the signal, and stationarity is assumed for the single segments. These values for the window length provide a good compromise between locality and efficiency of the coding scheme; the fixed-size constraint, however, clearly fails to identify and to exploit the particular characteristics of the signal under analysis. This problem has been addressed before in conjunction with speech coding techniques (see for example [6]); in the case of LPC coding in particular, the LPC coefficients for a given segment are obtained from the estimated autocorrelation function and, since segments are coded independently, the values for this estimate depend only on the local windowed data. It is clear that a better accuracy can be achieved by using windows dynamically adjusted in both length and positioning; segmentations of the data which extracts segments sharing the same statistics would then yield the optimal autocorrelation estimates. Furthermore, the overall quality of the global prediction for a given coding rate would improve if the order of the LPC predictor, the quantizers associated with the coefficients, and the coding method for the residuals were all adapting variables with respect to the segmentation process. The algorithm described in the previous sections can be applied to the LPC coding problem by letting $P$ be a set of meaningful combinations of predictor order, quantizers, and residual coding techniques and by finding the optimal segmentation with respect to the distortion measure determined by the elements of $P$.

In order to test the feasibility of this scheme we considered the well-known Government Standard Linear Predictive Coder LPC-10 [7]. This coder offers an output bitrate of 2400 bps and reasonably good subjective quality; speech is segmented in 22.5 ms frames (180 samples), the vocal tract characteristics are extracted by a order-10 LPC predictor, and the residual information is coded on a frame-by-frame basis by a voiced/unvoiced detector and a pitch extractor. Some dynamic programming techniques are already employed in the standard LPC-10 coder to allow for pitch tracking across frames and to position the analysis window in a semi-pitch-synchronous way; the span of the adaptation is however only three frames long. The dynamic LPC coder described in the reminder of this section utilizes the LPC-10 algorithm to analyze and encode the speech residual but replaces the fixed-size windowing by globally optimal dynamic segmentation and resource allocation.

With respect to the notation introduced in the previous sections, in this case the cell size is 90 sample long, half the size of the LPC-10 frame length, and $P$ is defined as the set of LPC predictors of order 6 to 10; quantization of the LPC coefficients takes place according to the LPC-10 specifications, namely 5 bits for the first four coefficients, 4 bits for the next four, 3 and 2 bits for the remaining two. Figure 2 shows the R/D curve obtained from sweeping $\lambda$ from zero to some maximum positive value for the minimization problem described by (3); in this case $R$ is the number of bits used to code the predictor's taps, and $D$ the corresponding MSE for the linear prediction solution. The values for $D$ are efficiently obtained as a by-product of Levinson's recursion while solving the linear prediction problem for all possible orders for a given segment; while not a measure of the "true" distortion introduced by the LPC coder, they can be seen as an overall goodness-of-fit index for the linear model. The operating points on the optimal R/D curve can be compared to the operating point of the standard LPC-10 coder, represented by the star in Figure 2.

Figure 3 shows the segmentations and model order selection relative to points A and B on the R/D curve. The plots show how the minimal data cells (the ticks on the x-axis) are joined together into segments, and which LPC order is selected for each segment. Point A illustrates how the algorithm can redistribute the same resources used by the fixed LPC to obtain a 0.3 dB decrease in the LPC MSE; acoustically, however, this is hardly a noticeable difference, given the already synthetic sound of LPC coded speech. More interestingly, point B shows how the data rate can be more than halved while keeping the same "distortion" for the LPC modelization.

The computational load of the global optimization algorithm increases quadratically with the length of the speech segment, as shown previously. Practically, however, long-range dependencies in typical speech signals span only a reasonably limited time interval; in terms of dynamic segmentation this means that, as the analysis progresses, the segmentation already delineated for the earlier part of the input signal is not likely to change anymore. The starting point for the optimization can therefore be advanced as soon as the backtracking algorithm detects that a sufficiently stable initial segmentation has been determined. This is exemplified in Figure 4; the plot shows how the segmentation evolves in time, with the fist-iteration segmentation at the bottom and the current segmentation at the top.

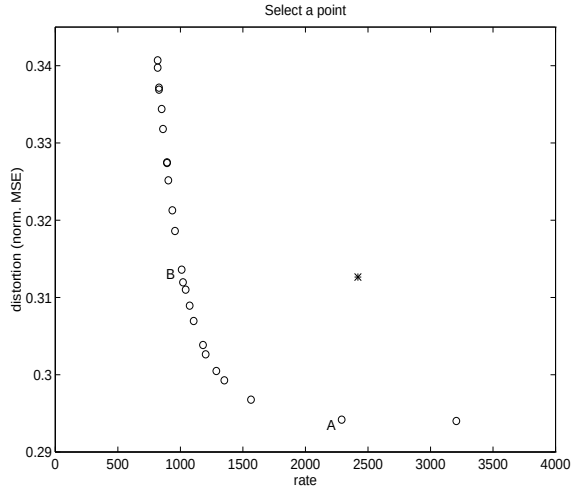


Figure 2. R/D curve for a simple segmentation example. The star represents the sub-optimal performance of a fixed-window LPC scheme.

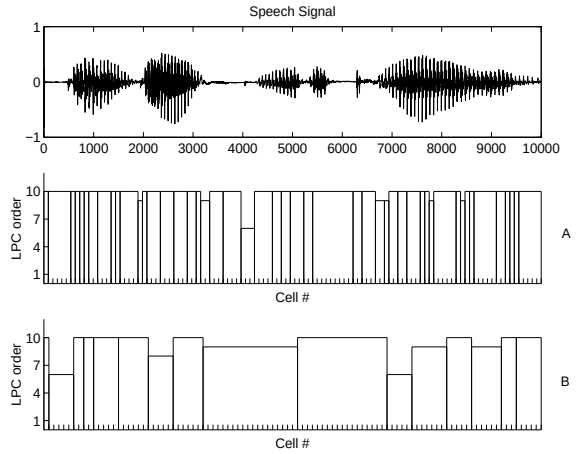


Figure 3. Speech signal (top box) and two segmentation/resource allocation results corresponding to operating points A (center box) and B (lower box) in the R/D curve of Figure 2.

The vertical axis corresponds to the iteration number for the algorithm: it is clear that as the algorithm progresses, the earlier segments are not modified further. Stable points for restart can be heuristically identified with the straight vertical lines which appear in the plot.

## 5. SINUSOIDAL MODELING

The proposed dynamic segmentation algorithm can be used in conjunction with any signal model where an additive cost such as a reconstruction error can be associated with application of the model to segments of the signal. A variety of signal representations are viable in this framework. We now consider the sinusoidal model, for which dynamic time segmentation offers immediate advantages over the static approach.

In sinusoidal modeling, the signal is modeled as a sum of nonstationary sinusoids or *partials*:

$$x(t) = \sum_{q=1}^{Q} A_q(t) cos\Theta_q(t)$$

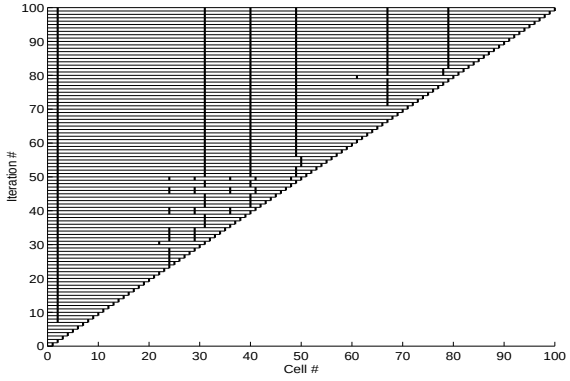


Figure 4. Evolution of the segmentation over time; the first segmentation (one cell only) is at the bottom, the current segmentation at the top.

Estimation of the model parameters is typically carried out using the short-time Fourier transform (STFT) with a fixed analysis frame size and a fixed stride between frames; a typical stride is half the frame size. The sinusoids are extracted by peak-picking in the STFT magnitude spectrum [8]. Such an analysis yields a frame-rate representation of the amplitude $A_q(t)$ and total phase $\Theta_q(t)$ of the constituent partials; in synthesis, these parameters are used to drive a bank of oscillators whose outputs are accumulated in accordance with the signal model. Because of its simplicity and flexibility, this representation in terms of sinusoidal parameters has proven effective for applications in speech coding and audio analysis-transformation-synthesis [8, 9]. The method, however, does have significant drawbacks for representing transient signals such as the attack of a musical note; this is explained by the following consideration of the sinusoidal synthesis algorithm.

As mentioned, sinusoidal synthesis is carried out using a bank of oscillators driven by amplitude and total phase control functions. Two difficulties arise in deriving these functions: line tracking and parameter interpolation; both arise because of the time-variation of the partials and the resultant frame-to-frame differences in the sinusoidal parameters. Since the analysis does not track the partials, but instead merely derives sets of parameters for the partials that it finds in the signal frames, the synthesis must establish continuity by relating the parameter sets in adjacent frames to form partials that endure in time. This line tracking is generally done by associating the $q$-th partial in frame $i$ to the partial in frame $i+1$ with frequency closest to $\omega_{q,i}$; this procedure is carried out until all of the partials in adjacent frames are either coupled or accounted for as a birth or a death, *i.e.* a partial that is newly entering or leaving the signal. After partial continuity is established by line tracking, it is necessary to interpolate the frame-rate sinusoidal parameters to derive the sample-rate oscillator control functions. This interpolation is typically done using low-order polynomial models such as linear amplitude and cubic total phase; these models are constrained to meet amplitude, frequency, and phase-matching criteria at the synthesis frame boundaries [8], which are separated in time by the analysis stride.

Fundamentally, the fixed-stride STFT analysis results in a delocalization of transient events such as attacks. This delocalization can also be interpreted in terms of the synthesis: the signal is reconstructed in each synthesis frame as a sum of linear-amplitude, cubic-phase sinusoids. Each of these sinusoids has the same time support, namely the synthesis frame size; this results in a smearing of signal features across the frame. In addition to this delocalization

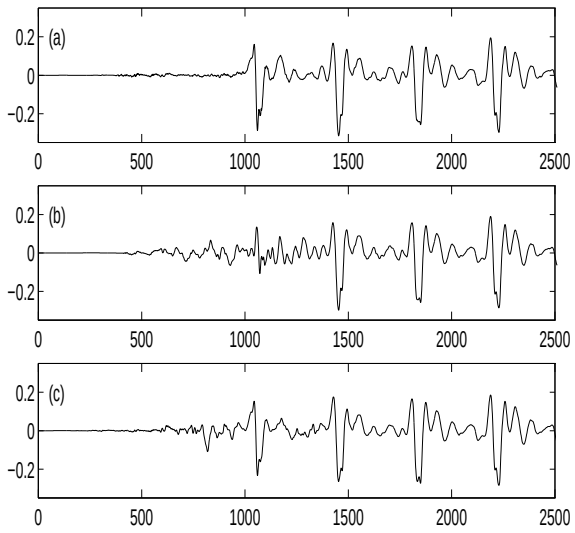


Figure 5. (a) A saxophone attack, (b) its delocalized reconstruction using a fixed frame size (512), and (c) a more accurate reconstruction using dynamic segmentation with the set of sizes $\{128, 256, 384, 512\}$; the algorithm chooses sizes 256 and 384 for the first two frames, and 512 for the rest.

within each frame, features are spread across neighboring frames by the line tracking and parameter interpolation operations. One consequence of this is a distortion of signal attacks, an example of which is shown in figure 5(b).

The distortion of transients can be reduced by using the proposed dynamic segmentation algorithm. In this approach, a set of synthesis frame sizes are allowed; given this set, the optimal segmentation, or time-varying frame size, is determined. In the optimization, the cost function for a segment is the mean-squared reconstruction error in the segment. Note that for a given segmentation, the analysis, as in the static case, is carried out using windows twice as wide as the synthesis segments; these analysis windows are centered at the synthesis segment boundaries. Because each segmentation requires a different set of analysis windows to cover the signal, each segmentation has its own set of sinusoidal analysis results. These various analyses are efficiently managed in the dynamic algorithm. Note however that because the sinusoidal parameters for a given synthesis segment are derived using analysis windows that extend outside the segment, the reconstruction error measure is not strictly independent from segment to segment. This dependence implies that the dynamic algorithm may not always find the optimal segmentation; however, simulations suggest that the dependence does not significantly interfere with the performance of the algorithm.

Fundamentally, the advantage of dynamic segmentation in the sinusoidal model is that the time support of the constituent linear-amplitude cubic-phase sinusoidal functions is adapted such that time-localized signal features are accurately represented. An example of the the distortion improvement is given in figure 5(c); the dynamic algorithm chooses shorter frame sizes near the attack to reduce delocalization. Another advantage is that the algorithm is able to choose long segments for regions where the signal does not exhibit transient behavior, thus improving the frequency resolution and the coding efficiency over the static case.

## REFERENCES

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

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

[3] C. Herley, J. Kovačević, K. Ramchandran, and M. Vetterli. Tilings of the time-frequency plane: Construction of arbitrary orthogonal bases and fast tiling algorithms. *IEEE Tran. on SP*, 41(12):3341–3359, December 1993.

[4] 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.

[5] N. S. Jayant and P. Noll. *Digital Coding of Waveforms.* Prentice-Hall, Englewood Cliffs, NJ, 1984.

[6] E. Bryan George, A. V. McCree, and V. R. Viswanathan. Variable frame rate parameter encoding via adaptive frame selection using dynamic programming. In *Proc. ICASSP*, volume 1, pages 271–274. IEEE, 1996.

[7] Thomas E. Tremain. The government standard linear predictive coding algorithm: Lpc-10. *Speech Technology*, pages 40–49, April 1982.

[8] R. McAulay and T. Quatieri. Speech analysis / synthesis based on a sinusoidal representation. *IEEE Tran. on ASSP*, 34(4):744–754, August 1986.

[9] X. Serra and J. Smith. Spectral modeling synthesis: A sound analysis / synthesis system based on a deterministic plus stochastic decomposition. *Computer Music Journal*, 14(4):12–24, Winter 1990.