

# Identifying Metrical and Temporal Structure with an Autocorrelation Phase Matrix \*

**Douglas Eck**

University of Montreal  
Department of Computer Science  
CP 6128, Succ. Centre-Ville  
Montreal, Quebec H3C 3J7 CANADA  
douglas.eck@umontreal.ca

## Abstract

This paper introduces a new method for detecting long-timescale structure in music. We describe a way to compute autocorrelation such that the distribution of energy in phase space is preserved in a matrix. The resulting Autocorrelation Phase Matrix (APM) is useful for several tasks involving metrical structure. In this paper we describe the details of calculating the APM. We then show how phase-related regularities from music are stored in the APM and present two ways to recover these regularities. The simpler approach uses variance or entropy calculated on the distribution of information in the APM. The more complex approach explicitly searches through the phase and lag space of the APM to predict meter and tempo in parallel. We compare these approaches against standard autocorrelation for the task of tempo prediction on a relatively large database of annotated digital audio files. We demonstrate that better tempo prediction is achieved by exploiting the phase-related information in the APM. We argue that the APM is an effective data structure for tempo prediction and related applications, such as real-time beat induction and music analysis.

Meter is the sense of stressed and unstressed beats that arises from the interaction among hierarchical levels of sequences having nested periodic components. Such a hierarchy is implied in Western music notation, where different levels are indicated by kinds of notes (whole notes, half notes, quarter notes, etc.) and where bars establish measures of an equal number of beats (Handel, 1993). Knowing the meter of a piece of music helps in predicting other components of musical structure such as the location of chord changes and repetition boundaries (Cooper and Meyer, 1960).

There have been many attempts to build computational models that attempt to discover periodic and metrical structure in digital audio. Oscillator models (Large and Kolen, 1994; Eck, 2002) have shown promise, though they have been used primarily with sequences of pulses or clicks (Large and Jones, 1999) rather than digital audio. Multi-agent systems such as those by Dixon (2001) have been applied with success to beat and meter-related tasks. Inference-based methods such as Monte-Carlo sampling (Cemgil and Kappen, 2003) and Kalman filtering (Cemgil et al., 2001) have also been used with success. However, the computational requirements for these methods can be prohibitively high.

In this paper we introduce the Autocorrelation Phase Matrix (APM), a two-dimensional data structure computed from MIDI or digital audio that stores information for estimating period and

---

\*Draft of version to appear in *Music Perception* 24(2):167-176, 2006.

phase at different levels in the metrical hierarchy. The main goal of this paper is to introduce details of the APM and show how it can be used to solve problems related to temporal structure in music, such as tempo prediction, meter prediction and beat induction. We provide algorithm details on how to compute the APM. We then introduce two ways to take advantage of the APM. The first, APM-VARIANCE, uses a variance measure of lag energy in the APM. The second, APM-PHASE, performs a lag-based search whereby phase-aligned elements in the matrix reinforce one another.

The structure of this paper is as follows. In Section 1 we define autocorrelation and discuss why it is a useful tool for detecting temporal structure in music. We also provide a short overview of existing music processing models that use autocorrelation or related methods. In Section 2 we provide mathematical and algorithmic details on how to compute the APM. In Section 3 we discuss three methods for predicting tempo from an audio file, one a baseline method (AUTOCORRELATION) and two others using the APM (APM-VARIANCE and APM-PHASE). In Section 4 we provide experimental results on a relatively large annotated dataset of audio files. These results indicate that APM-PHASE is a competitive model for tempo prediction, partially supporting the claim that the APM is a good data structure for music analysis and information retrieval tasks that rely on long-timescale structure in music. Finally, in Section 5 we present our conclusions and ideas for future research. Some of the simulation results discussed here were first presented at the ISMIR 2005 conference (Eck and Casagrande, 2005). However this is the first paper to provide a detailed description of the Autocorrelation Phase Matrix and the first introduction of the APM-PHASE approach.

## 1 Autocorrelation

The standard definition of autocorrelation is the cross-correlation of a signal with itself. The autocorrelation of a music signal generally reveals spikes corresponding to repetition in the signal. Autocorrelation can be computed via a series of summed dot-products of a signal with its shifted copy. The amount of shift is called the *lag* of the autocorrelation. The formula for the lag  $k$  autocorrelation  $a(k)$  for signal  $x$  having length  $N$  is

$$a(k) = \frac{1}{N} \sum_{n=k}^{N-1} x(n) x(n-k) \quad (1)$$

In general a range of lags within the rhythmical range (e.g.  $k \in [100\text{ms}, 101\text{ms}, \dots, 4000\text{ms}]$ ) would be computed. For music applications, these lags are proportionally long with respect to the length of a song or song segment. Thus *unbiased* autocorrelation is used

$$a'(k) = \frac{1}{N-k} \sum_{n=k}^{N-1} x(n) x(n-k) \quad (2)$$

where  $1/(N-k)$  is a normalization factor that accounts for the “shrinking” of available signal due to overlap as  $k$  grows.

Autocorrelation can also be defined in terms of Fourier analysis (Kunt, 1986):

$$a = \mathcal{F}^{-1}(|\mathcal{F}(x)|) \quad (3)$$

where  $\mathcal{F}$  is the Fourier transform,  $\mathcal{F}^{-1}$  is the inverse Fourier transform and  $||$  indicates taking magnitude from a complex value. In other words, autocorrelation is computed by performing

a time-to-frequency transform, discarding phase, and then moving back into the time domain. Observe that although the Fourier transform is reversible ( $X = \mathcal{F}^{-1}(\mathcal{F}(X))$ ) autocorrelation is irreversible due to the loss of phase.

This formulation reveals a relationship between autocorrelation and cepstral analysis. In fact, the cepstrum  $c$  is simply a log-scaled autocorrelation:

$$c = \mathcal{F}^{-1}(\log|\mathcal{F}(X)|) \quad (4)$$

The cepstrum is commonly used in speech analysis to separate the vocal tract transfer function (low frequency) from vocal excitation (high frequency). Our use of autocorrelation is similar to this source filter model approach with two differences. First, we are concerned only with the most salient repeating structure and so omit the log-scaling. This allows us to more easily identify salient lags. Second, we do not model the musical equivalent of the vocal excitation component.

Autocorrelation is also similar to comb filtering. A comb filter  $y(n) = \alpha x(n) + \beta x(n - k) + \kappa y(n - k)$  is an IIR filter with regularly-spaced spikes in its frequency response. Thus comb filters are excited by inputs having periodic structure. The main difference is that comb filters are additive over time while autocorrelation performs correlation over time using multiplication. A comb-filtered version of the APM is easy to create and was tested on a tempo tracking database. It offered no performance improvement over the autocorrelation model, a finding consistent with the observations of Klapuri et al. (2006).

## 1.1 Uses of autocorrelation and related methods in music

Space does not permit an exhaustive review of relevant approaches to rhythm and music. For one recent overview see Gouyon (2005).

Brown (1993) used autocorrelation to find meter in musical scores represented as note onsets weighted by their duration. Brown reported that the model was able to provide a reliable estimate of meter using relatively little computational power. The durational accent strategy she used is applicable for musical score analysis but is impractical for digital audio due to difficulties in reliably computing note durations.

Vos et al. (1994) proposed a similar autocorrelation method. The primary difference between their work and that of Brown was that Vos et al. used melodic intervals in computing accents. They applied their model to compositions by Bach, demonstrating the usefulness of melodic accent in detecting meter in these examples.

Scheirer (1998) used comb filters in a beat tracker for digital audio that performs relatively well over a range of musical styles (41 correct of 60 examples). He filtered an audio signal into several bands and then downsampled, differentiated and rectified each band. He then passed these signals into a bank of 150 comb filters, selecting the maximum output to recover the tempo and phase. Tempo changes were handled by repeatedly changing the choice of filter.

Klapuri et al. (2006) incorporated the signal processing approaches of Goto (2001) and Scheirer (1998) in a comb filter model that analyzed the period and phase of three levels of the metrical hierarchy. A Hidden Markov Model (HMM) was used for joint estimation of pulses at the three levels. The model was constrained using prior knowledge about human tapping rates and tempo change rates. The model won the ISMIR 2004 Tempo Induction contest (Gouyon et al., 2006). We return to this in Section 4.

Toiviainen and Eerola (2006) presented an autocorrelation-based model for meter induction on MIDI files. Their focus was on the relative usefulness of durational accent and melodic accent in predicting meter. The authors observed that durational and melodic accents provide a modest

boost in performance when used in conjunction with unaccented data, but that unaccented data was the most useful single factor for successful meter classification. They used stepwise discriminant function analysis to classify the meter of Finnish folk tunes (Eerola and Toiviainen, 2004) and European folk tunes (Schaffrath, 1995) using autocorrelation.

## 2 The Autocorrelation Phase Matrix (APM)

In this section we introduce the Autocorrelation Phase Matrix (APM), a data structure designed to address two limitations in current autocorrelation-based methods. First, autocorrelation is unable to track the phase relationship between signal and lag energy. Thus if phase estimation is necessary (as in beat tracking and other synchronization tasks) it must be handled separately from period estimation. The APM stores phase and period information in the same data structure, making it possible to incorporate both kinds of information in the same search (see Section 3.2).

Second, autocorrelation does not work well on non-percussive signals or noisy signals. (This weakness is shared by other methods such as comb filters and cepstral analysis). We address this in two ways. First we apply statistical measures of spread like variance or entropy to the information stored for each lag. The intuition behind this strategy is that musical repetition is generally stable in phase space. Thus at lags where repetition takes place, the APM should be relatively spiky. Statistics like entropy or variance are sensitive to this spikiness (see APM-VARIANCE in Section 3.2). Second we search the APM for combinations of hierarchically-related phase-aligned elements. This approach takes advantage of the fact that music often contains nested, phase-aligned temporal structure as seen in meter (see APM-PHASE in Section 3.2).

In describing the APM, we start with an overview of how standard autocorrelation can be used to find repeating temporal structure in music. Like most approaches, the first step of our algorithm is to extract from the audio signal a downsampled, rectified envelope of the signal. We tested our model on several existing envelope extraction algorithms, including that of Scheirer (1998). See Bello et al. (2005) for an overview. The one presented here worked better, although none of the approaches we tried yielded large improvements in performance. We computed the the log (dB) magnitude of a 1024-point spectrogram with log spacing of frequencies. The frequency band were then differentiated over time and summed, yielding an envelope sampled at 100Hz. Figure 1 shows the signal, the envelope and the autocorrelation of the envelope. In the figure the actual tempo of the song (484ms; 124 BPM) is marked with a vertical line. This tempo and its integer multiples are also marked with stars. Levels in the metrical hierarchy (periodicity of quarter note, half note, etc.) are marked with triangles.

The APM is an extension of an autocorrelation. For each lag  $k$  of interest, the APM stores intermediate results of autocorrelation in a vector of length  $k$  such that the results of the dot product from the autocorrelation are “wrapped” into that vector by their phase ( $\phi$ ). In this way, the APM (here denoted as  $P$ ) preserves the distribution of autocorrelation energy in phase space. At the same time, a counter matrix  $C$  allows for the computation of unbiased autocorrelation:

$$P(k, \phi) = \sum_{i=0}^{N/k-1} x(ki + \phi)x(k(i+1) + \phi) \quad (5)$$

$$C(k, \phi) = \sum_{i=0}^{N/k-1} 1 \quad (6)$$

For applications such as beat induction, it is useful to have a causal model so that processing can be

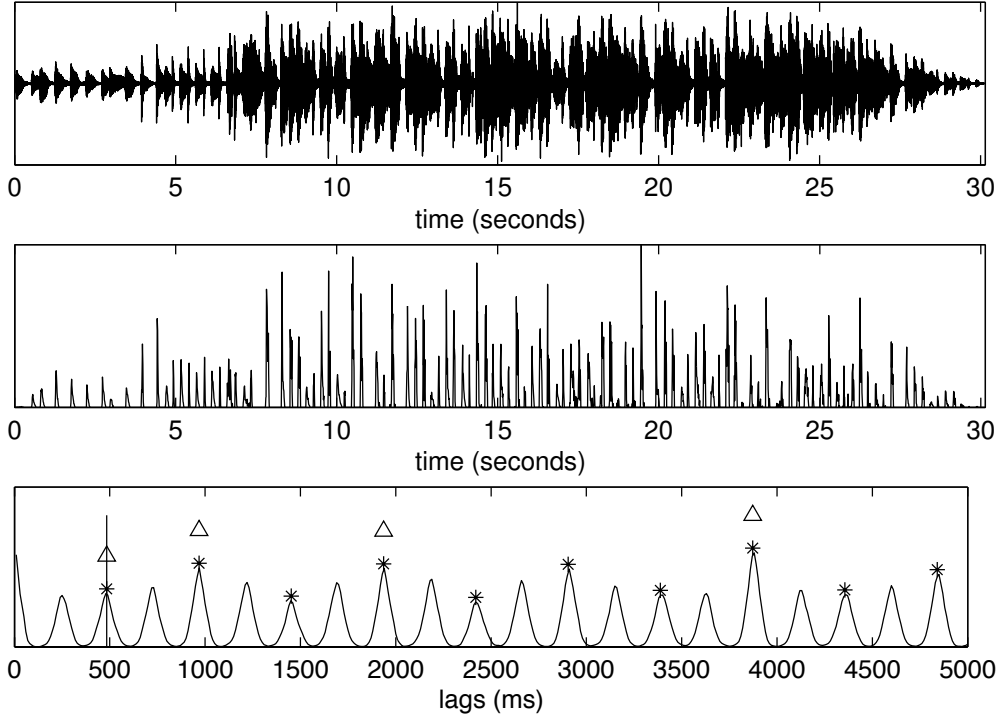


Figure 1: Timeseries (top), envelope (middle) and autocorrelation (bottom) of a ChaChaCha from the ISMIR 2004 Tempo Induction contest (Albums-Cafe\_Paradiso-08.wav). A vertical line marks the actual tempo (484 msec, 124bpm). Stars mark the tempo and its integer multiples. Triangles mark levels in the metrical hierarchy. In this example, tempo clearly aligns with autocorrelation spikes.

done online. The pseudocode in Algorithm 1 describes one simple causal version of the APM. Our own Matlab/C++ implementation is implemented using an optimized version of this algorithm<sup>1</sup>.

The key idea behind the APM is its ability to reveal repeating phase-correlated structure in a signal. This can be seen in the row-wise repetition of structure (Figure 2), which was computed from the same ChaChaCha song used above.

Autocorrelation  $a(k)$  and unbiased autocorrelation  $a'(k)$  can be recovered from the APM by summing all phase values for each lag, where the unbiased version is normalized using counter matrix  $C$ :

$$a(k) = \sum_{i=0}^{k-1} P(k, i) \quad (7)$$

$$a'(k) = \sum_{i=0}^{k-1} P(k, i)/C(k, i) \quad (8)$$

As already mentioned, it can also be useful to compute statistical measures of spread on the APM. The motivation for using these measures is that metrically-salient lags will tend to have more “spike-like” distribution due to the inherent periodicity of music. Thus even when the *autocorrelation* is relatively flat over all lags (as in the case of non-percussive instruments such as violins), the *distribution* of autocorrelation energy in phase space is often less flat, making it easier

<sup>1</sup>Code is available on request from the author.

---

**Algorithm 1** Pseudocode for computing APM.

---

**Input:**  $X$  {signal of length  $n$ }**Input:**  $K$  {set of lags of length  $m$ }**Input:**  $P, C$  {APM, APM counter matrix of size  $[m, n]$ }

```
1: for  $t \leftarrow 0$  to  $n - 1$  do
2:   for  $i \leftarrow 0$  to  $m - 1$  do
3:      $\phi \leftarrow \text{mod}(t, K[i])$ 
4:      $P[i, \phi] \leftarrow P[i, \phi] + X[t] * X[t + K[i]]$ 
5:      $C[i, \phi] \leftarrow C[i, \phi] + 1$ 
6:   end for
7: end for
```

---

to identify tempo and other musical structure. We consider two measures, entropy and variance. Entropy  $h$  is used to measure the amount of disorder in a system and is defined as

$$h(x) = - \sum_{i=0}^{N-1} x(i) \log_2[x(i)] \quad (9)$$

where  $x$  is a probability density. To compute entropy  $h(k)$  on the APM we normalize using two terms. First we divide  $P$  by  $C$  to yield unbiased autocorrelation. We then divide by  $a'(k)$  in order to force each row of the APM to sum to 1, thus allowing it to be treated as a probability density.

$$h(k) = \sum_{i=0}^{k-1} \frac{P(k, i)}{C(k, i) a'(k)} \log_2 \left[ \frac{P(k, i)}{C(k, i) a'(k)} \right] \quad (10)$$

In (Eck and Casagrande, 2005) we demonstrated that entropy yields significant performance improvements for tempo prediction over standard autocorrelation.

In the current study we consider (unbiased) variance (APM-VARIANCE, Section 3.2):

$$v(k) = \frac{1}{k-1} \sum_{i=0}^{k-1} \frac{(P(k, i) - \overline{P(k)})^2}{C(k, i)} \quad (11)$$

We use variance here primarily because, unlike entropy, it requires no normalization. Though it is conceptually useful to treat a row in the APM as a probability density, the necessary normalization yields degraded performance when very noisy signals are encountered.

### 3 Tempo prediction

Autocorrelation methods are unable to account for the non-uniform distribution of preferred musical tempo (i.e. some rates are more comfortable to play musically than others). For example, in Figure 1 the biggest autocorrelation value is seen at 3872ms. This value may indeed be the rate where the most repeating structure is observed, but it is far from the correct tempo, which is exactly 8 times faster. Tempo preferences can be accounted for by weighing the result of autocorrelation with a smooth function that is highest near the preferred production rate. The exact rate is somewhat disputed, although most agree that it lies somewhere between 500ms and 600ms IOI. We do not concern ourselves with this debate; see Van Noorden and Moelants (1999) for an overview.

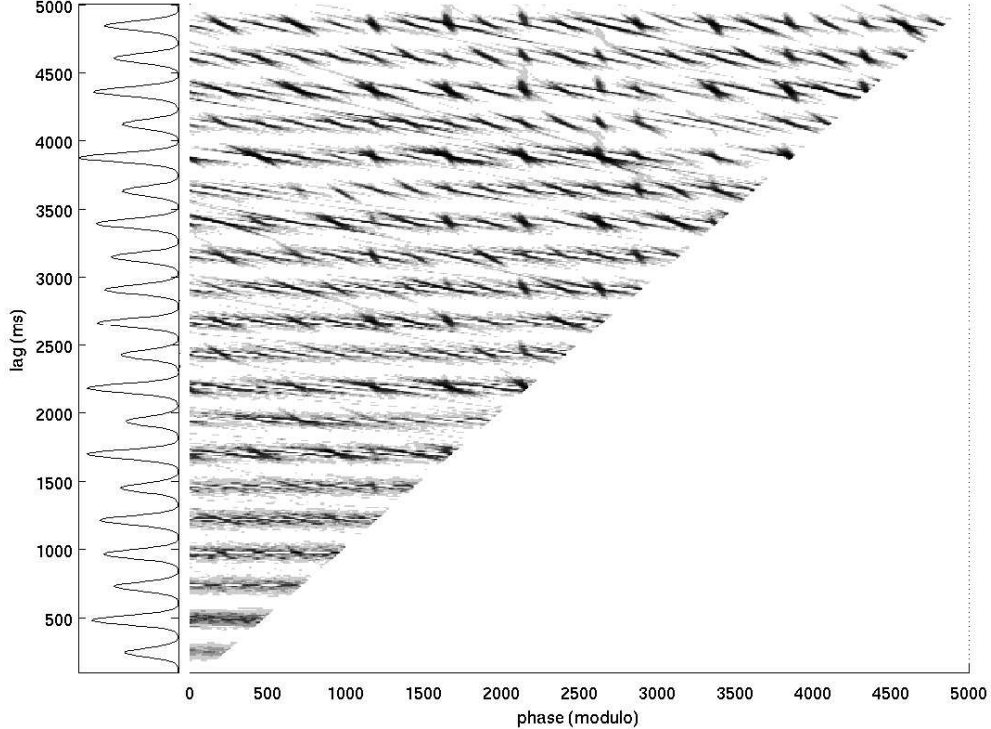


Figure 2: The APM for Albums-Cafe\_Paradiso-08.wav, the same song as shown in Figure 1. On the left the autocorrelation is recovered by summing rows in the matrix.

### 3.1 Tempo preference windows

In the following experiments, we will work with two existing tempo preference windows. The first window  $w_{Gauss}$  is taken from (Parncutt, 1994). It is based on the assumption that the distribution of preferred tempo is normal over log period and unimodal. A Gaussian model with two parameters, the standard deviation  $\sigma$  (recommended value 0.2) and the center of the distribution  $\mu$  in milliseconds (recommended value 600ms) is defined as

$$w_{Gauss} = \exp\left(\frac{1}{2}\left(\frac{1}{\sigma} \log_{10}\left[\frac{k}{\mu}\right]\right)^2\right) \quad (12)$$

where  $k$  is the predicted lag in milliseconds.

The second window  $w_{Res}$  is a resonance curve taken from (Van Noorden and Moelants, 1999). The resonance curve is based on the assumption that the distribution of preferred tempo is related to a damped resonating oscillator in the motor system. The details of this interesting theory are not considered here; see the source paper for details. The model has two parameters, a damping value  $\beta$  (recommended value 2.0) and a center frequency  $f$  (recommended value 2hz).

$$w_{Res} = \frac{1}{\text{sqrt}(f_0^2 - f_{ext}^2)^2 + \beta f_{ext}^2} - \frac{1}{\sqrt{f_0^4 + f_{ext}^4}} \quad (13)$$

These curves are shown below with experimental data in Figure 3.

## 3.2 Models

### The Autocorrelation baseline model

The tempo of a musical waveform can be estimated from the unbiased autocorrelation (Equation 8) by choosing the maximum magnitude lag  $k_A^*$  from among all lags. This yields the AUTOCORRELATION model

$$k_A^* = \operatorname{argmax}_k a'(k) w(k) \quad (14)$$

where  $w$  is one of the tempo preference windows  $w_{Gauss}$  or  $w_{Res}$ . Unless otherwise stated,  $w_{Gauss}$  was used for all experiments. See Section 4.2 for a comparison.

### The APM-Variance model

Tempo estimation is often more accurate when the variance (Equation 11) or entropy (Equation 10) of the APM phase information is used in place of the autocorrelation for each lag. This yields the APM-VARIANCE model.

$$k_V^* = \operatorname{argmax}_k v(k) w(k) \quad (15)$$

### The APM-Phase model

For tasks like tempo prediction, better results can often be obtained by integrating evidence at multiple metrically-related timescales. In previous work (Eck and Casagrande, 2005; Eck, 2004) we proposed a simple method for estimating in parallel the tempo and meter of a piece of music based on the summed values of several metrically-related lags. For example, the likelihood of a particular meter being in the music at tempo (lag)  $k$  is estimated as:

$$m(k) = \sum_{i \in \{1, 3, 6, 12\}} a'(ik) w(ik) \quad (16)$$

In this case the meter is triple, as can be seen by examining the values of  $i$ , which correspond to the hierarchal structure of a  $\frac{3}{4}$  meter. The winning tempo for that meter is simply the highest value  $k_M^*$  in  $m$  (compare to Equation 14):

$$k_M^* = \operatorname{argmax}_k m(k) \quad (17)$$

As an example of this process, see Figure 1. The triangles mark lags corresponding to a duple metrical hierarchy with  $k_M^*$  equal to 484ms. The model correctly predicts both the meter and the tempo for this example.

Any meter can be constructed using appropriate combinations of lags. Provided that the same number of points are used for all candidate meters, an overall winner can be chosen by selecting among the magnitudes  $m(k_M^*)$  for each meter. This search is efficient, requiring only a few additions per lag per meter.

In Eck (2004) we demonstrated that this method is good for predicting the meter of a song. In that study we used two annotated MIDI databases, the Essen Database (Schaffrath, 1995) of European folk melodies and the Finnish Database (Eerola and Toiviainen, 2004) of Finnish folk melodies. These datasets were also investigated by Toiviainen and Eerola (2006). From both

databases we selected melodies having either duple or triple/compound meters. This resulted in a total of 12646 songs of which 70% were duple. The model described in Equation 16 correctly classified 91% of the examples correctly, an accuracy rate competitive with the discriminant function analysis model from Toiviainen and Eerola (2006). See Eck (2004) for details.

It is possible to extend the strategy used in Equation 16 to take advantage of the APM. The idea is to search for winning combinations of aligned lag and phase values in the APM. The results of this phase-space search can be used to “align” the APM with the music sequence, thus performing beat induction. Although this approach can be generalized to any meter of interest, we will consider again only the triple case.

$$M(k, \phi) = \sum_{i \in \{1, 2, 4, 8\}} \sum_{j=0}^i P(ki, \phi j) w(ki) \quad (18)$$

where matrix  $M$  (smaller than the APM matrix  $P$ ) integrates phase-aligned points from a set of longer lags. Compare to Equation 11. The winning lag and phase combination  $\langle k_P^*, \phi^* \rangle$  is computed by finding the point of maximum value in this matrix. This yields the APM-PHASE model.

$$\langle k_P^*, \phi^* \rangle = \underset{(k, \phi)}{\operatorname{argmax}} M(k, \phi) \quad (19)$$

As in the simpler case (Equation 17), different candidate meters can be compared via their maximum values. Finally, because both lag *and* phase are inferred by APM-PHASE, it is possible to use these results to perform beat induction. An in-depth beat induction study is currently under preparation.

When tempo is fixed, long musical segments can be treated with a single APM using the method described above. However, when tempo changes over time, the resulting APM becomes “smeared” or “blurred”. To address this we computed individual APMs on overlapping short segments of audio. For our experiments we used a segment length of 5 seconds overlapped by 4 seconds, thus yielding a new meter and tempo prediction each second, starting at 5 seconds into the song. These segment-level results were combined by choosing the most frequently-predicted tempo and meter as a global winner.

## 4 Experiments

In this section we describe experiments that use the APM to predict the mean tempo of an audio file. We present two extensions to this baseline that gradually increase both the complexity and performance of the model. The first extension is to use a measure of the spikiness of each phase entry entropy or variance to sharpen the  $a'(k)$ . The second extension is to infer tempo and meter in parallel and to incorporate evidence at multiple levels of the winning metrical hierarchy. In this way we consider the influence of the metrical hierarchy on tempo selection by incorporating evidence from longer-timescale correlation.

### 4.1 Setup

We used two datasets from the ISMIR 2004 Tempo Induction contest (Gouyon et al., 2006). The first dataset was called “Ballroom” and consisted of 697 wav files each approximately 30 seconds in duration encompassing eight genres. The second dataset was called “Song Excerpts”. This dataset

Table 1: Overall tempo prediction results (ISMIR 2004 contest data)

Model	Ballroom		Song Excerpts		Both	
	Acc. A	Acc. B	Acc. A	Acc. B	Acc. A	Acc. B
AUTOCORRELATION	56%	85%	53%	73%	55%	80%
APM-VARIANCE	55%	89%	52%	81%	54%	86%
APM-PHASE	<b>65%</b>	<b>94%</b>	<b>64%</b>	87%	<b>65%</b>	<b>91%</b>
KLAPURI	63%	91%	58%	<b>91%</b>	61%	<b>91%</b>

consisted of 465 songs of roughly 20sec duration spanning nine genres. A third database “Loops” was not available after the contest due to copyright restrictions. We had not yet developed this model at the time of the contest and so did not compete.

The contest calculated two accuracy measures. “Acc. A” is the number of predictions within 4% of the target tempo. “Acc. B” is the number of predictions within 4% of the target tempo multiplied by 1, 2, 3, 1/2 or 1/3, thus allowing for mismatches based on choosing the wrong level in the metrical hierarchy. So that we can compare our results with those of the contest, we use the same error measures here

We use all three models described in Section 3. We designed our experiments such that the baseline AUTOCORRELATION model takes no advantage of the APM. The APM-VARIANCE model benefits from the APM somewhat (in that spiky APM entries receive higher value). The APM-PHASE model takes greater advantage of the APM, performing an explicit search through metrically-related, phase-aligned entries in the matrix in order to select a winning meter.

## 4.2 Results

The overall results are very promising. Standard autocorrelation yielded 80% correct overall for “Accuracy B”. The APM-VARIANCE model yielded 86% and the APM-PHASE model yielded 91% correct, thus supporting our claim that phase information as stored in the APM is useful for tempo prediction. In analyzing errors we observed that several files had short periods of missing data. These sounded like clicks when played on a loudspeaker. They were not handled well by the model because our envelope computation took the derivative of the spectrogram, which was excessively large for these errors. Given that these sorts of errors are not uncommon when compact discs are read into a computer, this represents a weakness in our envelope computation. Second, (presumably to make the contest hard) the “Song Excerpts” database contained at least a few songs having uncommon meters such as  $\frac{5}{4}$ . APM-PHASE sometimes assigned tempo that was either  $\frac{4}{5}$ th or  $\frac{5}{4}$ th of the target tempo, resulting in a error. This could be dealt with by considering more meters in the search performed in Equation 19. However it is not clear whether these additional meters would degrade performance for other songs. In Table 1 we report error as percentage for each dataset separately as well as a summary for both. For comparison, the results for the contest winner (Klapuri et al., 2006) are shown as KLAPURI .

### Tempo preference windows

In Figure 3, three tempo preference curves are shown. See Section 3.1 for details on the curves and their parameters. For the plot, we centered the curves at 500ms in keeping with Van Noorden and Moelants (1999). However, the value 600ms works better for this dataset, as can be seen in Table 2.

Table 2: Success rates (percent) of tempo preference curves for APM Phase model

Tempo preference curve	Ballroom		S.Excerpts		Both	
	Acc. A	Acc. B	Acc. A	Acc. B	Acc. A	Acc. B
None	45%	92%	50%	<b>87%</b>	47%	90%
Resonance ( $\beta = 1.12; 500\text{ms}$ )	61%	88%	49%	82%	56%	86%
Resonance ( $\beta = 1.12; 600\text{ms}$ )	<b>65%</b>	<b>94%</b>	62%	84%	64%	90%
Resonance ( $\beta = 2.0; 600\text{ms}$ )	<b>65%</b>	93%	60%	85%	63%	90%
Gaussian ( $\sigma = 0.2; 600\text{ms}$ )	<b>65%</b>	<b>94%</b>	<b>64%</b>	<b>87%</b>	<b>65%</b>	<b>91%</b>

It is clear that the curve improve performance, especially for “Acc. A” (compare 47% correct for no window versus 65% correct for the Gaussian window). However the effects for “Acc. B” are not so pronounced (90% without a window versus 91% with Gaussian). This disparity is not so surprising given that the “Acc. B” measure considers faster and slower tempos as correct. Given the relatively small number of songs in the dataset ( $N = 1161$ ) and the fact that data, especially Ballroom, is not representative of music in general, we cannot conclude much from these results about which window is best.

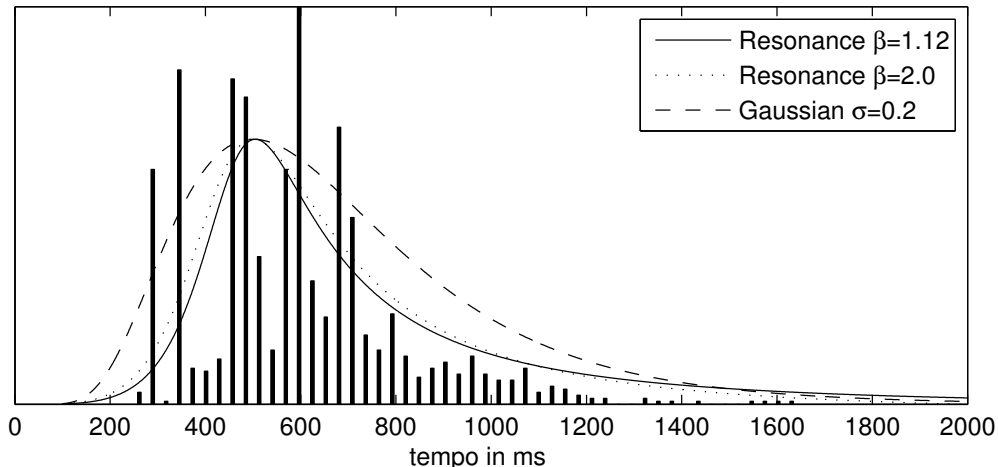


Figure 3: Three windows for implementing tempo preferences. All windows are centered at 500ms. A 50-bin histogram of labeled tempos from Ballroom and Excerpts is shown for comparison. See text for further details.

In Table 3 we see a breakdown of performance for APM-PHASE by genre for the Ballroom dataset. Space constraints prohibit reporting similar results for “Song Excerpts”. It can be seen that failure in “Acc. A” is often style-specific. One possible explanation is that the tempos of certain styles are not near the 600ms center of the tempo preference window. This hypothesis is supported by the strong correlation between the distance of mean tempo from 600ms and failure on “Acc. A”.

Table 3: Performance of APM-PHASE by genre on the Ballroom dataset. See text for details.

Style	Count	Acc. A	Acc. B	Mean Tempo
ChaChaCha	111	96%	97%	489ms
Jive	60	5%	98%	364ms
Quickstep	82	0%	94%	299ms
Rumba	97	96%	93%	603ms
Samba	86	93%	97%	594ms
Tango	86	93%	98%	473ms
VienneseWaltz	65	0%	97%	337ms
Waltz	110	81%	81%	699ms

### 4.3 Discussion

Overall the APM Phase model performed at a level that is competitive with the winning model of Klapuri et al. from the contest. However it is worth noting that the Klapuri model performed better on the “Acc. B” measure for the Song Excerpts dataset. In our opinion, the Song Excerpts dataset is the more challenging and more diverse of the two datasets. Thus we cannot conclude that our approach is superior. More tests on much larger labeled datasets are required.

The major difference and, we believe, the main success of the KLAPURI model is their use of an HMM for incorporating information from three levels of the metrical hierarchy and their use of the dynamic programming technique of Viterbi alignment to find an optimal state through the HMM lattice. (An explanation of HMMs and Viterbi alignment is not possible given space constraints. See Klapuri et al. (2006) for all details.) The APM is not computationally expensive provided a reasonable limit is placed on the number of lags. Specifically its time complexity is  $O(kt)$  where  $k$  is the number of lags in the autocorrelation and  $t$  is the length of the signal.

## 5 Future Work and Conclusions

We are pursuing two main directions in our current research. First we are working on a real-time online version that predicts tempo and beat in music. The extensions to the work presented here to handle online beat tracking are fairly minor. One issue is that of adapting the APM to changing tempo. The strategy used here of computing multiple segments and voting on their predictions is not applicable to online beat tracking. We are investigating methods similar to the HMM/Viterbi strategy used by Klapuri et al. (2006). Second we are interested in how the APM can be used in the context of music analysis and automatic composition (Eck and Schmidhuber, 2002). These tasks are hard in part because it is difficult to correlate events over long timescales. The APM provides evidence about which timescales are important, thus constraining the task. Finally we are investigating better methods for envelope computation, a crucial step in this process. One interesting direction of inquiry concerns the use of supervised machine learning methods to enhance the detection of salient note onsets. See Lacoste and Eck (2006) for more.

In this paper we introduced an autocorrelation-based method for detecting long-timescale structure in music. The Autocorrelation Phase Matrix (APM) stores intermediate results of autocorrelation in a way that preserves the phase relationship between autocorrelation energy and the signal. We described details of how the APM works, provided an algorithm (here called APM-

PHASE) that searches the APM in parallel for tempo and meter, preserving both the periodicity and phase relationship to the signal for the winning meter. The model was shown to perform very competitively on a tempo induction dataset consisting of 1162 digital audio files of varying style. We believe that these results demonstrate the utility of the APM, and that more research in this direction is warranted.

## References

- Bello, J., Daudet, L., Abdallah, S., Duxbury, C., Davies, M., and Sandler, M. (2005). A tutorial on onset detection in music signals. *IEEE Transactions on Speech and Audio Processing*, 13:1035–1047.
- Brown, J. (1993). Determination of meter of musical scores by autocorrelation. *Journal of the Acoustical Society of America*, 94:953–957.
- Cemgil, A. T. and Kappen, H. J. (2003). Monte Carlo methods for tempo tracking and rhythm quantization. *Journal of Artificial Intelligence Research*, 18:45–81.
- Cemgil, A. T., Kappen, H. J., Desain, P., and Honing, H. (2001). On tempo tracking: Tempogram representation and Kalman filtering. *Journal of New Music Research*, 28(4):259–273.
- Cooper, G. and Meyer, L. B. (1960). *The Rhythmic Structure of Music*. The Univ. of Chicago Press, Chicago.
- Dixon, S. E. (2001). Automatic extraction of tempo and beat from expressive performances. *Journal of New Music Research*, 30(1):39–58.
- Eck, D. (2002). Finding downbeats with a relaxation oscillator. *Psychol. Research*, 66(1):18–25.
- Eck, D. (2004). A machine-learning approach to musical sequence induction that uses autocorrelation to bridge long timelags. In Lipscomb, S., Ashley, R., Gjerdingen, R., and Webster, P., editors, *The Proceedings of the Eighth International Conference on Music Perception and Cognition (ICMPC8)*, Adelaide. Causal Productions.
- Eck, D. and Casagrande, N. (2005). Finding meter in music using an autocorrelation phase matrix and shannon entropy. In *Proceedings of the 6th International Conference on Music Information Retrieval (ISMIR 2005)*, pages 504–509, London: University of London.
- Eck, D. and Schmidhuber, J. (2002). Finding temporal structure in music: Blues improvisation with LSTM recurrent networks. In Bourlard, H., editor, *Neural Networks for Signal Processing XII, Proceedings of the 2002 IEEE Workshop*, pages 747–756, New York. IEEE.
- Eerola, T. and Toiviainen, P. (2004). Digital Archive of Finnish Folktunes. Computer database. University of Jyväskylä.
- Goto, M. (2001). An audio-based real-time beat tracking system for music with or without drum-sounds. *Journal of New Music Research*, 30(2):159–171.
- Gouyon, F. (2005). *A Computational Approach to Rhythm Description*. PhD thesis, Department of Technology of the University Pompeu Fabra, Barcelona, Spain.

- Gouyon, F., Klapuri, A., Dixon, S., Alonso, M., Tzanetakis, G., Uhle, C., and Cano, P. (2006). An experimental comparison of audio tempo induction algorithms. *IEEE Transactions on Speech and Audio Processing*. (In press).
- Handel, S. (1993). *Listening: An introduction to the perception of auditory events*. MIT Press, Cambridge, Mass.
- Klapuri, A., Eronen, A., and Astola, J. (2006). Analysis of the meter of acoustic musical signals. *IEEE Trans. Speech and Audio Processing*, 14(1):342–355.
- Kunt, M. (1986). *Digital Signal Processing*. Artech House, Norwood, Mass.
- Lacoste, A. and Eck, D. (2006). A supervised classification algorithm for note onset detection. *EURASIP Journal on Applied Signal Processing*. (In press).
- Large, E. W. and Jones, M. R. (1999). The dynamics of attending: How people track time-varying events. *Psychological Review*, 106(1):119–159.
- Large, E. W. and Kolen, J. F. (1994). Resonance and the perception of musical meter. *Connection Science*, 6:177–208.
- Parncutt, R. (1994). A perceptual model of pulse salience and metrical accent in musical rhythms. *Music Perception*, 11:409–464.
- Schaffrath, H. (1995). The Essen Folksong Collection in Kern Format. Computer database. Center for Computer Assisted Research in the Humanities.
- Scheirer, E. (1998). Tempo and beat analysis of acoustic musical signals. *Journal of the Acoustical Society of America*, 103(1):588–601.
- Toiviainen, P. and Eerola, T. (2006). Autocorrelation in meter induction: The role of accent structure. *Journal of the Acoustical Society of America*, 119(2):1164–1170.
- Van Noorden, L. and Moelants, D. (1999). Resonance in the perception of musical pulse. *Journal of New Music Research*, 28(1):43–66.
- Vos, P., van Dijk, A., and Schomaker, L. (1994). Melodic cues for metre. *Perception*, 23:965–976.