A Tempo-Extraction Algorithm Using an Autocorrelation Phase Matrix and Shannon Entropy

Douglas Eck
University of Montreal
Department of Computer Science
CP 6128, Succ. Centre-Ville
Montreal, Quebec H3C 3J7 CANADA
eckdoug@iro.umontreal.ca

ABSTRACT

Our algorithm is based on autocorrelation. What distinguishes it from other autocorrelation approaches is that we computes the distribution of lag autocorrelation energy as a function of phase. This results in a lag-by-phase matrix that compactly represents the repetitive structure of a musical example. The model is designed to allow meter analysis of audio files. However it can also be used to extract information about tempo, as demonstrated in this MIREX entry. To compute tempo, we enhance a standard autocorrelation of the signal with the row-wise Shannon entropy of the phase information in the matrix. This enhances the resolution of the autocorrelation, in many cases allowing for the detection of tempo in musical examples where autocorrelation does not work.

Keywords: autocorrelation, autocorrelation phase matrix, entropy

1 Introduction

We describe a model that uses autocorrelation as its core, but that takes advantage of the distribution of energy in phase space as a method to overcome weaknesses in standard autocorrelation. To sum, the model works as follows:

- Preprocess audio or MIDI file to yield envelope sampled at or near 1000Hz
- Compute autocorrelation while preserving the distribution of energy in phase space
- Multiply the autocorrelation for each lag by the entropy of the energy in phase space for each lag.
- For meter prediction, integrate phase entropy over a number of hierarchically-related lags constructed to match a particular metrical interpretation. Choose the winning hierarchy
- For tempo induction report the lag value from the winning set of lags that is closest to a comfortable tapping rate
- For predicting phase, for the lags in the winning hierarchy search the autocorrelation phase matrix bottom-up (from fastest to slowest lag) for the most salient phase. Use phase energy from faster lags to constrain the choice at slower lags. Report the selected phase at slowest lag.

The model is described in Sections 1.1 through 1.5. For a longer description of the model see Eck (2005), available on the author's website at http://www.iro.umontreal.ca/~eckdoug/publications.html

1.1 Preprocessing

For MIDI files, the onsets can be transformed into spikes with amplitude proportional to their midi note onset volume. Alternately MIDI files can simply be rendered as audio and written to wave files. Stereo audio files are converted to mono by taking the mean of the two channels. Then files are downsampled to some rate near 1000Hz. The actual rate is kept variable because it depends on the original sampling rate. For CD-audio (44.1Khz), we used a sampling rate of 1050Hz allowing us to downsample by a factor of 42 from the original file. Best results were achieved by computing a sum-of-squares envelope over windows of size 42 with 5 points of overlap. However for most audio sources a simple decimation and rectification works as well. The model was not very sensitive to changes in sampling rate nor to minor adjustments in the envelope computation such as substituting RMS (root mean square) for the sum of squares computation.

One of our goals was to avoid preprocessing as much as possible, and we succeeded in doing so. However there is no reason that our model could not be adapted to work with a multi-band filtering approach similar to, e.g., Klapuri et al. (2005); Goto (2001).

1.2 Autocorrelation Phase Matrix

The method of cross-correlation is commonly used to evaluate whether two signals exhibit common features and are therefore correlated (Ifeachor and Jervis, 1993). To perform cross-correlation one computes the sum of the products of corresponding pairs of two signals. A range of lags are considered, accounting for potential time delays between correlated information in the two signals. The formula for the lag k cross-correlation C_k between sig-

nals x_1 and x_2 (having length N) is:

$$C_k(X_1, X_2) \quad = \quad \frac{1}{N} \sum_{0<n<N-k} x_1(n) * x_2(n+k) \quad (1)$$

Autocorrelation is a special case of cross-correlation where $x_1 == x_2$. There is a strong and somewhat surprising link between autocorrelation and the Fourier transform. Namely the autocorrelation A of a signal X (having length N) is:

$$A(X) \quad = \quad ifft(|fft(X)|) \quad (2)$$

where fft is the (fast) Fourier transform, $ifft$ is the inverse (fast) Fourier transform and $||$ is the complex modulus. One advantage of autocorrelation for our purposes is that it is defined over periods rather than frequencies (note the application of the IFFT in Equation 2), yielding better representation of low-frequency information than is possible with the FFT.

Autocorrelation values for a random signal should be roughly equal across lags. Spikes in an autocorrelation indicate temporal order in a signal, making it possible to use autocorrelation to find the periods at which high correlation exists in a signal. As a music example, consider the autocorrelation for a ChaChaCha from the ISMIR 2004 Tempo Induction contest is shown (Figure 1). The peaks of the autocorrelation align with the tempo and integer multiples of the tempo.

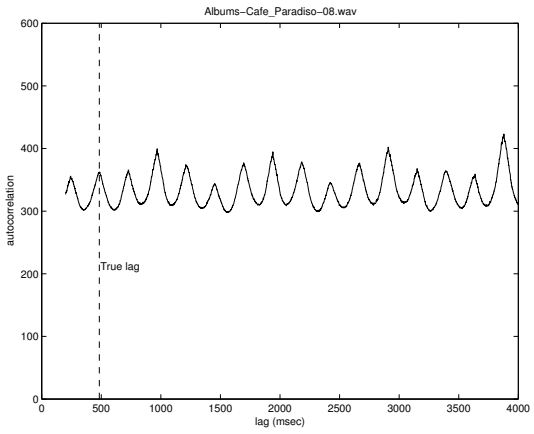

Figure 1: Autocorrelation of a ChaChaCha from the ISMIR 2004 Tempo Induction contest (Albums-Cafe_Paradiso-08.wav). The dotted vertical line marks the actual tempo of the song (484 msec, 124 bpm).

Unfortunately autocorrelation has been shown in practice to not work well for many kinds of music. For example when a signal lacks strong onset energy, as it might for voice or smoothly changing musical instruments like strings, the autocorrelation tends to be flat. See for example a song from Manos Xatzidakis from the ISMIR 2004 Tempo Induction in Figure 2. Here the peaks are less sharp and are not well-aligned with the target tempo. Note that the y-axis scale of this graph is identical to that in Figure 1.

One way to address this is to apply the autocorrelation to a number of band-pass filtered versions of the signal, as

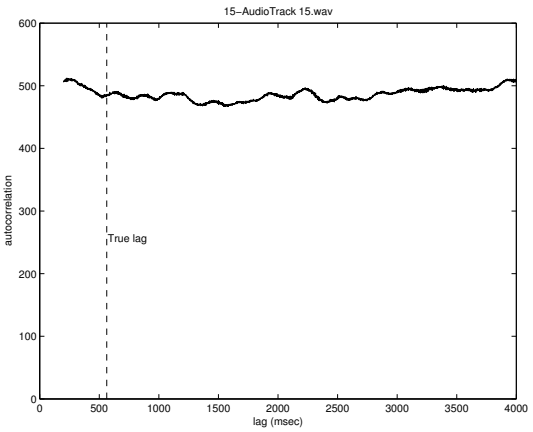

Figure 2: Autocorrelation of a song by Manos Xatzidakis from the ISMIR 2004 Tempo Induction contest (15-AudioTrack 15.wav). The dotted vertical line marks the actual tempo of the song (563 msec, 106.6 bpm). Compare the flatness of the autocorrelation and the lack of alignment between peaks and the target.

discussed in Section 1.1. In place of multi-band processing we compute the distribution of autocorrelation energy in phase space. This has a sharpening effect, allowing autocorrelation to be applied to a wider range of signals than autocorrelation alone without extensive preprocessing.

The autocorrelation phase information for lag l is a vector A_l:

$$A_l = \left(\sum_{i=0}^{\lfloor \frac{N-l}{l} \rfloor} x_{li+\phi} \, x_{l(i+1)+\phi} \right)_{\phi=0}^{l-1} \quad (3)$$

We compute an autocorrelation phase vector A_l for each lag of interest. In our case the minimum lag of interest was 200ms and the maximum lag of interest was 3999ms. Lags were sampled at 1ms intervals yielding $L = 3800$ lags. Equation 3 effectively "wraps" the signal modulo the lag l question, yielding vectors of differing lengths ($|A_l| == l$). To simplify later computations we normalized the length of all vectors by computing a histogram estimate. This was achieved by fixing the number of phase points for all lags at K ($K = 50$ for all simulations; larger values were tried and yielded similar results but significantly smaller values resulted in a loss of temporal resolution) and resampling the variable length vectors to this fixed length. This process yielded a rectangular autocorrelation phase matrix P where $|P| = [L, K]$.

As an example of an autocorrelation phase table, consider Figure 3, which shows the rectified normalized signal from a piano rendition of one of the rhythmic patterns from Povel and Essens (1985). The pattern was rendered with a base inter-onset-interval of 300ms. On the left in Figure 4 the autocorrelation phase matrix is shown. On the right, the sum of the matrix is shown. It is the standard autocorrelation.

1.3 Autocorrelation Phase Entropy

As already discussed, is possible to improve significantly on the performance of autocorrelation by taking advan-

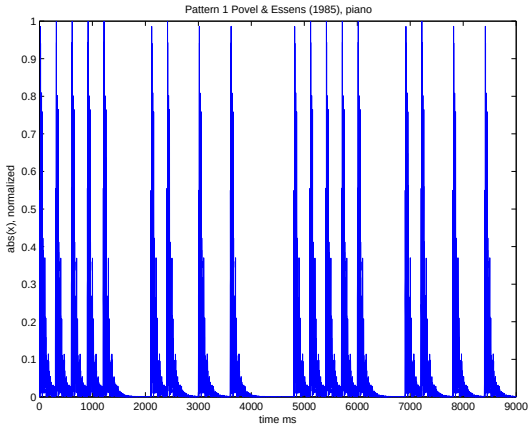

Figure 3: The rectified normalized signal generated by creating a piano rendering from a MIDI version of Povel & Essens Pattern 1. Two repetitions of the length-16 nine-event pattern are shown. See Povel and Essens (1985) for details.

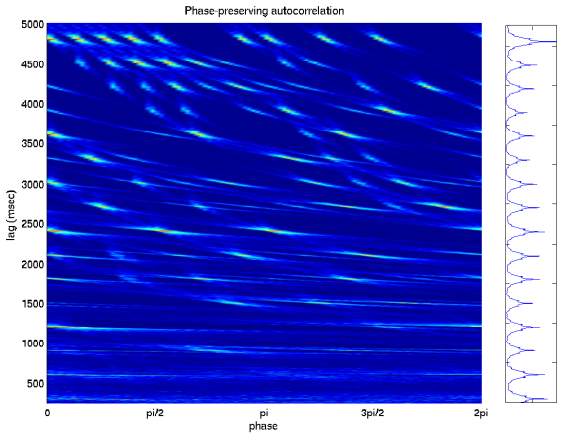

Figure 4: The autocorrelation phase matrix for Povel & Essens Pattern 1 computed for lags 250Ms through 500ms. The phase points are shown in terms of relative phase $(0, 2\pi)$. On the right it is shown that taking the sum of the matrix by row yields exactly the autocorrelation.

tage of the distribution of energy in the autocorrelation phase matrix. The idea is that metrically-salient lags will tend to be have more "spike-like" distribution than non-metrical lags. Thus even if the *autocorrelation* is evenly distributed by lag, the *distribution of autocorrelation energy in phase space* should not be so evenly distributed. There are at least two possible measures of "spikiness" in a signal, variance and entropy. We focus here on entropy, although experiments using variance yielded very similar results.

Entropy is the amount of "disorder" in a system. Shannon entropy H:

$$H(X) \quad = \quad -\sum_{i=1}^{N} X(i) log_2[X(i)] \qquad (4)$$

where X is a probability density. We compute the entropy for lag l in the autocorrelation phase matrix by as follows:

$$A_{sum} \quad = \quad \sum_{i=0}^{N} A_l(i) \qquad (5)$$

$$H_l \quad = \quad -\sum_{i=0}^{N} A_l(i)/A_{sum} log_2[A_l(i)/A_{sum}] \qquad (6)$$

This entropy value, when multiplied into the autocorrelation, significantly improves tempo induction. For example, in Figure 5 we show the autocorrelation along with the autocorrelation multiplied by the entropy for the same Manos Xatzidakis show in in Figure 2. On the bottom observe how the detrended (1- entropy) information aligns well with the target lag and its multiples. (Detrending was done to remove a linear trend that favors short lags. Simulations revealed that performance is only slightly degraded when detrending is omitted.) Most robust performance was achieved when autocorrelation and entropy were multiplied together. This was done by detrending both the autocorrelation and entropy vectors, scaling them both between 0 and 1 and then multiplying them together.

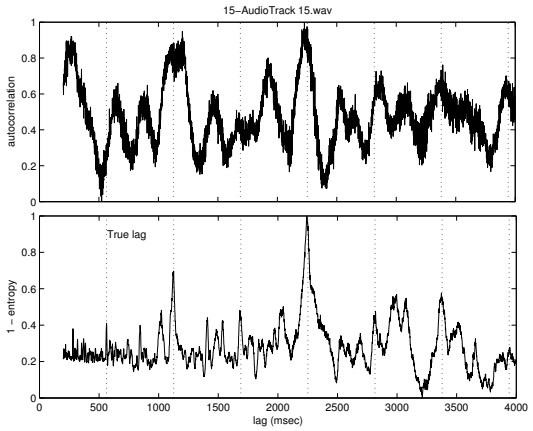

Figure 5: Autocorrelation and entropy calculations for the same Manos Zatzidakis song shown in Figure 2. The top is the autocorrelation and is identical to Figure 2 except that it is scaled to $[0, 1]$. On the bottom is (1 - entropy), scaled to $[0, 1]$ and detrended. Observe how the entropy spikes align well with the correct tempo lag of 563ms and with its integer multiples (shown as vertical dotted lines in both plots.

1.4 Metrical hierarchy selection

We now move away from the autocorrelation phase matrix for the moment and address task of selecting a winning metrical hierarchy. A rough estimate of meter can be had by simply summing hierarchical combinations of autocorrelation lags. In place of standard autocorrelation we use the product of autocorrelation and (1 - entropy) AE as described above. The likelihood of a duple meter M^{duple} existing at lag l can be estimated using the following sum:

$$M_l^{duple} = AE(l) + AE(2l) + AE(4l) + AE(8l) \qquad (7)$$

The likelihood of a triple meter is estimated using the following sum:

$$M_l^{triple} = AE(l) + AE(3l) + AE(6l) + AE(12l) \qquad (8)$$

Other candidate meters can be constructed. using similar combinations of lags. A winning meter can be chosen by sampling all reasonable lags (e.g. $200ms <= l <=$

$2000ms$) and comparing the resulting M_l^* values. Provided that the same number of points are used for all candidate meters, these M_l^* values can be compared directly, allowing for a single winning meter to be selected among all possible lags and all possible meters. Furthermore, this search is efficient given that each lag/candidate meter combination requires only a few additions.

1.5 Prediction of tempo

Once a metrical hierarchy is chosen, there are several simple methods for selecting a winning tempo from among the winning lags. One option is to pick the lag closest to a comfortable tapping rate, say 600ms. A second better option is to multiply the autocorrelation lags by a window such that more accent is placed on lags near a preferred tapping rate. The window can be applied either before or after choosing the hierarchy. If it is applied before selecting the metrical hierarchy, then the selection process is biased towards lags in the tapping range. We tried both approaches; applying the window before selection yields better results, but only marginally better (on the order of 1% better performance on the tempo prediction tasks described below). To avoid adding more parameters to our model we did not construct our own windowing function. Instead we used the function (with no changes to parameters) described in Parncutt (1994): a Gaussian window centered at 600ms and symmetrical in log-scale frequency.

2 Contest Results

The model placed 8th out of 13 entries with an overall score of 0.644 standard deviation. The model predicted at least one tempo correct for 86.43% of the songs. When we consider that the winning entry (M. Alonso) got 95% correct and that G. Peeters got 95.71% correct, our performance can be seen as respectable but certainly not stellar. When both tempos are considered, the model correctly predicted 53.57% of the songs. This is close in raw percentage to that of the winning entry (M. Alonso) value of 55.71%. However C. Uhle reported a score of 59.29%.

We were surprised to see that our model was competitive in terms of runtime. For example, the winning submission of M. Alonso returned a runtime of 2875 while ours was 1675. We submitted a highly-unoptimized matlab version. In order to keep runtimes reasonable we set the parameters of the model rather aggressively. It is possible that our model could have performed better if we had allowed it to compute a larger, higher-resolution Phase Autocorrelation Matrix. In any case, the runtimes are difficult to compare given that different machines were used for different models.

As has been discussed via an email list among competitors and organizers, it is difficult to know how close the models really are in terms of performance. We are convinced by these results that several models perform better than our model—and we congratulate Miguel Alonso for his winning entry— but it remains unclear how much better these models are. What is needed is a labeled dataset having at least one order of magnitude more files. Lacking that, it would be interesting to run these models on last years tempo induction contest as a comparison.

3 Conclusions

This paper introduces a novel way to detect metrical structure in a music and to use meter as an aid in detecting tempo. Post-contest results reveal that the model finished well in the middle of the pack among 13 contestants. There are clear ways to improve the algorithm. For example, multi-band processing could be used as a preprocessing step and the algorithm could be run on each band. Also, an online version of the model that makes predictions on a frame-by-frame basis and integrates evidence leads to better performance. An online version was implemented after the contest deadline and performs better than the submitted version on datasets used in our lab. However more research in this direction is warranted.

In our view one of the more interesting aspects of this model is that it provides a relatively compact representation of temporal structure based on phase. By computing standard Shannon entropy over phase values, we easily outperformed autocorrelation at tempo finding. We hope that more complex analyses will yield still better performance and will make it possible to use the Autocorrelation Phase Matrix in other domains such as score quantization and beat induction.

4 Acknowledgments

We wish to thank the Audio Tempo Extraction contest organizers Martin McKinney and Dirk Moelants for their initiative in conceiving and executing the contest. We also wish to thank J. Stephen Downie, Emmanuel Vincent and the rest of the MIREX 2005 team for all of their hard work.

References

Douglas Eck. Meter and autocorrelation. In *10th Rhythm Perception and Production Workshop (RPPW) 2005*, Alden Biesen, Belgium, 2005.

Masataka Goto. An audio-based real-time beat tracking system for music with or without drum-sounds. *Journal of New Music Research*, 30(2):159–171, 2001. URL `http://staff.aist.go.jp/m.goto/PAPER/JNMR2001goto.pdf`.

E. C. Ifeachor and B. W. Jervis. *Digital Signal Processing: A Practical Approach*. Addison-Wesley Publishing Company, 1993.

A. Klapuri, A. Eronen, and J. Astola. Analysis of the meter of acoustic musical signals. *IEEE Trans. Speech and Audio Processing*, 2005. To appear.

R. Parncutt. A perceptual model of pulse salience and metrical accent in musical rhythms. *Music Perception*, 11:409–464, 1994.

D.J Povel and Peter Essens. Perception of temporal patterns. *Music Perception*, 2:411–440, 1985.