# USING SOURCE SEPARATION TO IMPROVE TEMPO DETECTION

**Parag Chordia**
Dept. of Music, GTCMT
Georgia Institute of Technology

**Alex Rae**
Dept. of Music, GTCMT
Georgia Tech Center for Music Technology

## ABSTRACT

We describe a novel tempo estimation method based on decomposing musical audio into sources using principal latent component analysis (PLCA). The approach is motivated by the observation that in rhythmically complex music, some layers may be more rhythmically regular than the overall mix, thus facilitating tempo detection. Each excerpt was analyzed using PLCA and the resulting components were each tempo tracked using a standard autocorrelation-based algorithm. We describe several techniques for aggregating or choosing among the multiple estimates that result from this process to extract a global tempo estimate. The system was evaluated on the MIREX 2006 training database as well as a newly constructed database of rhythmically complex electronic music consisting of 27 examples (IDM DB). For these databases the algorithms improved accuracy by 10% (60% vs 50%) and 22.3% (48.2% vs. 25.9%) respectively. These preliminary results suggest that for some types of music, source-separation may lead to better tempo detection.

## 1. BACKGROUND AND MOTIVATION

A working definition of tempo is the rate of the underlying rhythmic pulse of music determined by a human listener tapping along to the music, typically expressed in beats per minute (BPM). This may differ from a notated tempo, and different listeners, or the same listener at different times, often entrain to different metrical levels, so that some tapping rates may be half or double as fast as others. Further, in some types of music, the most natural way to tap along is asymmetric (e.g. tapping on the accented first and third beat in a fast group of five beats). For our purposes, these complexities are important to acknowledge at the outset as they set natural bounds on performance and suggest appropriate ways of judging accuracy.

Tempo estimation is a fundamental MIR task and underlies almost all rhythmic descriptions of music. However, state-of-the-art tempo detection is still highly variable in its accuracy, working well on most simple cases, but often performing poorly or not at all on rhythmically complex music [1]. The current work is motivated by two observations: 1) rhythmically complex music may be constructed out of components or layers (e.g. musical parts or sources) that are rhythmically simpler than the mix and thus easier to track; 2) in many types of music, humans track the beat or the tempo by hearing out a particular instrument or part. For example, in many types of rhythmically complex electronic music, a "click track" is present in the mix. More generally, in many musical genres a particular part plays a time-keeping function: for example, in standard jazz the walking bass line is the time keeper, in Indian music the tabla, in Afro-Cuban music the clave. Being able to hear out these time-keeping parts makes tempo tracking easier for humans.

## 2. RELATED WORK

The starting point of the current work is tempo detection that looks for periodicities in the signal by taking the autocorrelation of the detection function (ACF). A good review of current algorithms can be found in McKinney et al. [2] as well as specific descriptions of autocorrelation-based approach in Ellis [3] and Davies and Plumbley [4]. Recent work has explored the extension of this basic approach to tempo detection in a variety of ways. Wright et al. [5] describe a system that searches for the rhythmic pattern of the clave in Afro-Cuban music and show that such an approach out-performs techniques more reliant on isochronous events such as the Ellis and Dixon [6] algorithms. In their work, a matched filter is used to extract the clave from the mix. In this paper, we attempt to generalize the idea of finding the time-keeper in the mix in a way that is less reliant on domain-specific knowledge. Seyerlehner et al. [7] cast tempo estimation as a nearest neighbor problem, representing instances using a smoothed autocorrelation function (ACF). This approach suggests the idea of using not just the peak of the ACF, but including other features to improve tempo detection. Xiao et al, [8] demonstrate that using timbral features in addition to ACF-based features can reduce double/half tempo errors and indicates that even very crude uses of timbre can improve tempo estimation accuracy. Earlier work on tempo detection has also sought to improve accuracy by processing information in particular frequency sub-bands [9, 10]. In some cases, this is akin to a crude source separation, for example, separating the bass drum from the rest of a song.

Probabilistic latent component analysis (PLCA), a technique for source-separation described in Section 3.2, has been used for unmixing as well as transcription [11, 12].

The closely related technique of non-negative matrix factorization (NMF) has been used to improve drum detection [13, 14]. In these works, tracks were separated into sources that were then grouped into either tonal or percussive layers based on features of the components. This is relevant to the current work because it demonstrates the idea of using source-separation as a pre-processing step to improve performance on a standard MIR task. Additionally, features of the components are used to classify them into different groups, a technique used in this work to judge how strong a pulse different components have.

## 3. METHOD

### 3.1 Overview

As stated, the technique described here builds on ACF-based tempo detection. First, the track is separated into components using a single-channel source separation method (PLCA). Next, the tempo of each component is estimated on the separated audio. The component tempo estimates, along with the windowed ACF that was used to calculate the component tempo, are then used to find a global tempo estimate for the excerpt. We discuss several attempts to solve the problem of finding the best tempo estimate from the components. Two basic strategies were employed: selecting the tempo of a component with the highest estimated rhythmic clarity, and clustering component tempo estimates and weighting each cluster by the rhythmic clarity of each element in the cluster. Figure 1 shows a block diagram of the system.

### 3.2 Source Separation

Blind source separation attempts to recover constituent elements from a signal without any specific *a priori* knowledge of their characteristics. For audio, this corresponds to "unmixing," the reconstruction of a clean signal of each of a number of sounds that have been mixed together. Faithful reconstruction of component elements has a wide array of potential applications; in the current work we are less interested in mimicking the timbre of the original sources than in capturing rhythmic characteristics that may be less evident in a full mix.

We approach this task using the non-shift-invariant version of Probabilistic Latent Component Analysis (PLCA) [11, 12]. The input to the PLCA is a spectrogram, computed using a 1024 sample Hann window with a hopsize of 256 samples and then normalized to be a valid probability distribution. Latent variables representing the components are estimated using expectation maximization, and the output consists of a magnitude spectrum and relative contribution over time for each component; the number of desired components must be specified by the user.

After some experimentation, we set the number of components to be extracted to eight. A more systematic evaluation of the optimal number components remains for future work. The corresponding timbral and temporal profiles were used to synthesize audio for each component using phase information from the original audio.

### 3.3 Tempo Estimation

The tempo was estimated for each component using the Ellis algorithm [3]. The algorithm constructs a detection function based on a 40-channel db-magnitude mel spectrogram. First the signal is downsampled to 8 kHz, mixed to mono and divided into 32 ms frames with a 4 ms hopsize. The first-order difference is taken for each channel, and the sum of positive values across all channels is the value of the detection function for that frame (spectral flux). The autocorrelation of the detection function is calculated and then windowed to bias it towards tempos close to 120 BPM. The windowing effectively excludes tempos falling outside an acceptable range, and at the same time mimics the natural preference of humans to tap at rates between 90-120 BPM [1]. The tempo estimate is simply the lag time corresponding the peak value of the windowed ACF, converted to BPM. Any peaks before the first zero-crossing of the ACF are disallowed to prevent spurious peaks near zero lag. A small modification was made to the Ellis algorithm so that the top ten tempo candidates were returned rather than a single best tempo estimate, defined as the BPMs corresponding to the ten highest peaks in the windowed ACF. These additional tempo estimates were used in the clustering method described below. For all other techniques, only the best estimate for each component was used.

Each component was tempo tracked in this way, resulting in ten candidate tempos for each component. This meant that for a given track there were 80 tempo candidates (8 components × 10 estimates). The ACF value associated with each candidate and the entire windowed ACF were also stored, and these were used to help select the best global tempo estimate from the candidates.

### 3.4 Tempo Selection

Below we describe several approaches to selecting a single tempo estimate from the candidates.

#### *3.4.1 Pulse-clarity*

Inspired by the idea that certain components might accurately represent a relatively isochronous part of the track, the first approach focused on finding the best component from which to estimate the global tempo. That is, we attempted to find the component with the clearest pulse, and then choose the highest ranked tempo estimate for that one component as the global tempo estimate.

Lartillot et al. [15] showed that several features of the ACF are correlated with human judgments of pulse-clarity. Intuitively, the idea is that a relatively isochronous part with clear onsets will lead to an ACF that has well-defined and relatively large peaks. Following Lartillot et al., we calculated the following features on the ACF: maximum, minimum, and kurtosis. Additionally we added entropy and sparseness [16] as features, with sparseness defined as:

$$\text{sparseness(x)} = \frac{\sqrt{n} - (\sum |x_i|)/\sqrt{\sum x_i^2}}{\sqrt{n} - 1} \tag{1}$$

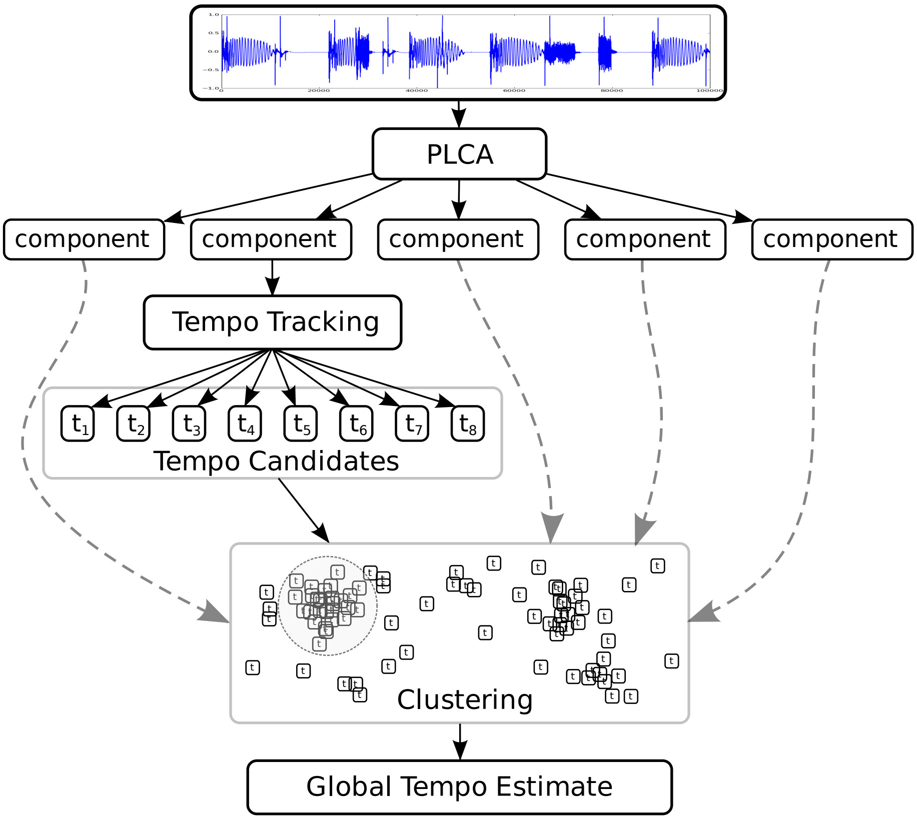

**Figure 1**. Block diagram of tempo estimation algorithm

Before calculating these features the ACF was normalized so that higher amplitude components would not dominate. For each component the ACF values were divided by the sum of the absolute value of all ACF values. The max and min were simply the maximum and minimum ACF values after normalization, and we expected that larger absolute values would correspond with greater pulse-clarity. Kurtosis was used to measure the peakiness of the ACF, i.e. how well-defined the ACF peaks were. Entropy and sparseness also assessed peakiness.

Each feature was evaluated separately; Table 2 summarizes the performance of each feature (evaluation criteria are discussed in Section 4.2). It can be seen that the most obvious feature, the maximum ACF value, outperformed the other measures on the IDM09 data, while they were about equal on the MIREX06 data.

In order to make better use of pulse-clarity features, an attempt was made to apply them to a more systematic supervised machine-learning framework. For this, we trained a multivariate Gaussian classifier using a ten-fold cross validation scheme. In addition to the ACF features we defined a new set of features based on the ratios of the candidate tempo estimates for each component. These features were based on the idea that we would expect to see harmonically related peaks in the ACF of rhythmically clear components. The ratio between every possible pairing of the ten candidate tempos was computed, leading to $\binom{10}{2}$, i.e. forty-five ratios per component. We then computed a histogram of these values in the range .45 to 2.05 with a bin width of .1, leading to 23 features. The targets were binary, representing whether the component estimate matched the ground truth. The tempo of an excerpt was calculated by choosing the tempo associated with the component that had the highest posterior probability, i.e. the greatest likelihood of its tempo matching the ground truth given the ACF features. This approach worked well for the MIREX06 data but less so for the IDM09 data (Table 2). At this point it is difficult to say whether the IDM09 performance was due to an insufficiently large training database to accurately learn the multivariate distribution or if more discriminative features must be found.

### 3.4.2 Clustering

Another method was attempted for determining the global tempo, based on the idea of taking a vote among the candidate tempos, possibly weighted by the corresponding normalized ACF values. The basic intuition was that the true tempo should appear more frequently than spurious estimates among the candidate tempos. To implement this, we first partitioned the candidate tempos into clusters using a hierarchical cluster tree. However, a simpler approach that did not attempt an exclusive partitioning performed better. In the latter approach, the candidate tempos for all components and their associated normalized ACF values were merged into a single matrix. For each tempo candidate, a score was determined by summing the ACF values for that tempo as well as for any tempos that were half or double, within a 5% tolerance. Of course such a method will often lead to ties, which we resolved by choosing the tempo closest to 120 BPM. Because we chose to measure accuracy accounting for half and double matches (see Section 4.2), this was not a major issue. The tempo candidate with the highest score was chosen as the global tempo estimate. To see if ACF weighting was important we also performed experiments ignoring ACF values and assigning scores by simply counting the number of elements in each cluster. However, ACF weighting consistently improved performance and was chosen as the default. Additionally, we experimented with multiplying each ACF value by the pulse-clarity estimate of the component based on the heuristics described above. This did not affect results and was therefore not included in the final version.

## 4. EVALUATION

### 4.1 Databases

Evaluation was performed on two databases. The main database consisted of twenty-seven 30-second excerpts chosen from the IDM/glitch genre of electronic music (IDM09), with an emphasis on tracks that we thought were rhythmically complex and layered. For each excerpt, two independent manual annotations were made. [1] For all excerpts the human annotators agreed, with the exception of a few half/double conflicts. In those cases, we randomly selected a single estimate. It should be noted that our accuracy measure allowed for half/double errors.

Additionally, the twenty publicly available MIREX06 training excerpts were used [2]. These consisted of a mix of genres and tempo ranges, and included annotation of two tempos representing the two highest peaks in a distribution of tempos calculated from listeners' tapping times. For our experiments we simply selected the ground truth tempo that was more commonly assigned.

### 4.2 Accuracy measure

We defined a match to be whenever the estimated tempo matched the annotated tempo, or double or half the annotated tempo, within a five percent tolerance window. Evaluation of tempo detection algorithms is somewhat dependent on the end-goal. We might reasonably hope that the tempo detection algorithm would correspond to judgments of human listeners. However, although there may be a fair degree of reliability between judgments for simple rhythms, there can be substantial disagreement about the appropriate metrical level or even the tempo for more rhythmically complex music. Moreover, more experienced listeners often tap at a lower metrical level (i.e. slower tempo) than novice listeners and in some cases novice listeners tap irregularly and are unable to clearly sense the tempo. Although this may be trivially true for music with no clear rhythm, this can also occur for music where there is a high degree of reliability for experienced listeners. For retrieval tasks, such as selecting tracks with similar tempos, it might be more appropriate to consider a match only when the metrical level of the main ground truth annotation is matched. On the other hand, for transcription or synchronization tasks it is appropriate to consider matches at different metrical levels. Because our emphasis here was on IDM, a genre that often contains metrical level ambiguity, we decided that this latter definition of accuracy made the most sense.

| | **Baseline (Ellis)** | **Clustering** | **Change** |
|---|---|---|---|
| **MIREX06** | 0.50 | 0.60 | 0.10 |
| **IDM09** | 0.26 | 0.48 | 0.22 |
| **combined** | 0.36 | 0.53 | 0.17 |

**Table 1**. The primary results are summarized here for each of the databases as well as for the combined set. The baseline is the Ellis algorithm run on the unseparated excerpts. Clustering refers to choosing the global estimate according to the procedure described in Section 3.4.2

### 4.3 Results

To get a sense of the upper-bound of performance for each track we checked to see if the true tempo was the primary tempo estimate for any of the components, and also whether the true tempo was present in any of the candidate tempos. Since subsequent steps attempt to filter these values, our pulse-clarity based technique can do no better than this first value, and the clustering method can do no better than the latter. The primary component tempo was correct for 70.4% of excerpts from IDM09 and 75% of MIREX06. A match was found in a candidate tempo of one of the components 96.3% and 85% of the time for IDM09 and MIREX06 respectively. Of course it should be noted while that we would expect this percentage to increase as the number of candidate tempos per component increases, the number of false positives will also tend to increase. Nevertheless these data suggest a high performance ceiling.

Table 1 summarizes the main the results, while Table 2 provides a more complete view of the performance of the different algorithms described in the paper. The first column in both tables is the baseline performance, given by running the Ellis algorithm on the unseparated excerpt using the definition of accuracy described above. Baseline accuracy for the MIREX06 data was 50% and 25.9% for IDM09. The substantially lower baseline accuracy for IDM09 reflects the rhythmic complexity of these excerpts. It can be seen that for the MIREX06, IDM09, and combined databases that the clustering algorithms improved accuracy by 10% (60% vs 50%) , 22.3% (48.2% vs. 25.9%) and 17% (53.2% vs. 36.2%), respectively. From the detailed results table we can see that the ML-based approach achieved a 10 percentage-point improvement on MIREX06 (60% vs 50%), and a 3.7 percentage-point increase on IDM09.

Clustering using multiple candidates per component, as well as the ML-based approach, had an accuracy of 60% for MIREX06, a 10% improvement on the baseline. In the case of IDM09 there was a substantial improvement of 22.3% (48.2% vs. 25.9%). However in this case the ML-based approach was only marginally better than baseline.

[1] The IDM09 database and the tempo annotations will be made publicly available online.

| | **Baseline** | **Pulse** | | | | | **Clustering** | **ML** | |
|---|---|---|---|---|---|---|---|---|---|
| | Ellis | min | max | entropy | kurtosis | sparseness | | all features | pulse-only |
| **MIREX06** | 0.50 | 0.35 | 0.40 | 0.40 | 0.40 | 0.40 | 0.60 | 0.60 | 0.60 |
| **IDM09** | 0.26 | 0.15 | 0.33 | 0.26 | 0.26 | 0.30 | 0.48 | 0.27 | 0.26 |
| **combined** | 0.36 | 0.23 | 0.36 | 0.32 | 0.32 | 0.34 | 0.53 | 0.43 | 0.40 |

**Table 2**. Detailed accuracy results for each of the pulse-clarity measures described in Section 3.4.1 as well as for the machine learning algorithm also described in Section 3.4.1. For the ML algorithm results are shown for all features, as well as with only the original pulse heuristic features.

For both data sets using pulse-clarity alone did not improve results, with the exception of the max ACF (33.3% vs. 25.9%) and sparseness (29.6% vs 25.9%) features for IDM09.

## 5. DISCUSSION AND CONCLUSION

From these data it seems that the clustering-based approach is the superior method, particularly when compared to using a single component as the basis for the global estimate. It is possible, however, that this is simply an artifact of an inaccurate source separation step. Auditioning components reveals that many components are not true sources at all but parts of sources or several sources; source separation is still a delicate art. Nevertheless, many components do clearly correspond to parts and at times one can clearly hear time-keeping parts popping out. This noisiness probably accounts for the fact that the clustering approach, which retains more information about possible periodicities by retaining multiple tempo estimates for each component, is more robust. Although the current work did not bear out the ML-based approach, we believe that systematic incorporation of multidimensional rhythmic information will play an important part in future component-based tempo detection algorithms.

We have shown that for these data, using source separation in conjunction with clustering can substantially improve results, particularly for rhythmically complex and layered material. We have also explored a variety of techniques for implementing the core idea of using source decomposition to improve tempo estimation. In particular, we developed techniques for tempo estimation based on pulse clarity scoring of components and clustering of component tempo estimates. As source separation techniques improve, it should be possible to more closely mimic the rhythmic perception of humans, which in many cases is based on recognizing distinct parts that have a time-keeping function.

We expect that the approach described here will be most useful for layered, rhythmically complex music that tends to have simpler sub-parts. For simpler music, on the other hand, the less dramatic results are unlikely to justify the computational cost of source separation. We expect that this method will fail for music where the rhythm is emergent, i.e. only becomes apparent when several layers are played simultaneously.

## 6. FUTURE WORK

There are many possible extensions to this work. Thus far we have done little work to tune the source separation step. For example, what is the optimum number of components? It is likely that eventually this should be set adaptively based on the characteristics of the piece and the likely number of sources. These, however, remain unsolved problems, though the recent surge in research on single-channel source separation using PLCA and NMF is likely to dramatically improve our unmixing capabilities. Additionally, we intend to pursue the ML-based approach. In the long-run, it is likely that some combination of features can be found that will determine more reliably whether a component tempo estimate is the correct global estimate. And, as always, only with the expansion of the tempo database, and additional benchmarking against multiple systems, will we truly be able to assess the strengths and weaknesses of the techniques presented here.

## 7. REFERENCES

[1] Martin F. McKinney and Dirk Moelants? Ambiguity in tempo perception: What draws listeners to different metrical levels? *Music Perception*, 24(2):155–166, 2006.

[2] M. F. Mckinney, D. Moelants, M. E. P. Davies, and A. Klapuri. Evaluation of audio beat tracking and music tempo extraction algorithms. *Journal of New Music Research*, 36(1):1–16, 2007.

[3] Daniel P. Ellis. Beat tracking by dynamic programming. *Journal of New Music Research*, 36(1):51–60, 2007.

[4] M. E. P. Davies and M. D. Plumbley. Beat tracking with a two state model. In *Proceedings of the IEEE International Conference on Acoustics, Speech and Signal Processing*, 2005.

[5] Matthew Wright, Andrew Schloss, and George Tzanetakis. Analyzing afro-cuban rhythm using rotation-aware clave template. In *Proceedings of International Conference on Music Information Retrieval*, 2008.

[6] Simon Dixon. Evaluation of the audio beat tracking system beatroot. *Journal of New Music Research*, 26(1):39–50, 2007.

[7] K. Seyerlehner, G. Widmer, and D. Schnitzer. From rhythm patterns to perceived tempo. In *Proceedings of International Conference on Music Information Retrieval*, 2007.

[8] Linxing Xiao, Aibo Tian, Wen Li, and Jie Zhou. Using a statistic model to capture the association between timbre and perceived tempo. In *Proceedings of International Conference on Music Information Retrieval*, 2008.

[9] Eric D. Scheirer. Tempo and beat analysis of acoustic musical signals. *The Journal of the Acoustical Society of America*, 103(1):588–601, 1998.

[10] 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.

[11] Paris Smaragdis, Bhiksha Raj, and Madhusudana Shashanka. Supervised and semi-supervised separation of sounds from single-channel mixtures. In *Proceedings of the 7th International Conference on Independent Component Analysis and Signal Separation*, London, UK, September 07.

[12] Paris Smaragdis and Bhiksha Raj. Shift-invariant probabilistic latent component analysis. Technical report, 2007.

[13] Christian Uhle, Christian Dittmar, and Thoma Sporer. Extraction of drum tracks from polyphonic audio using independent subspace analysis. In *4th International Symposium on Independent Component Analysis and Blind Signal Separation*, 2003.

[14] Marko Heln and Tuomas Virtanen. Separation of drums from polyphonic music using non-negative matrix factorization and support vector machine. In *In: Proc. EUSIPCO2005. (2005*, 2005.

[15] Olivier Lartillot, Tuomas Eerola, Petri Toiviainen, and Jose Fornari. Multi-feature modeling of pulse clarity: Design, validation, and optimization. In *Proceedings of International Conference on Music Information Retrieval*, 2008.

[16] Patrik Hoyer. Non-negative matrix factorization with sparseness constraints. *Journal of Machine Learning Research*, 5:1457–1469, 2004.