BEAT TRACKING WITH PARTICLE FILTERING ALGORITHMS

Stephen Hainsworth

Cambridge University Engineering Department
Cambridge CB2 1PZ, UK.
swh21@eng.cam.ac.uk

Malcolm Macleod

QinetiQ
Malvern WR14 3PS, UK.
m.macleod@signal.qinetiq.com

ABSTRACT

This paper presents a beat tracking algorithm for musical audio signals. The method firstly extracts musical changepoints from the help signal and then uses a particle filtering algorithm to associate these to a tempo process. Results are comparable with the current state of the art.

1. INTRODUCTION

This paper focuses upon one area of the broad subject which is musical audio analysis: beat tracking. The aim is to extract information about the meter or tempo of a musical signal; essentially build an algorithm which taps its hypothetical foot along to the music. This is a task which most humans are able to perform for many types of music so it is reasonable to expect that a computer can be made to replicate this activity. This is in comparison to say the full polyphonic transcription problem (e.g. [1]) which for humans requires years of musical training and will probably require a similar level of algorithmic complexity for a computer.

However, beat tracking is a low level task in the overall transcription problem and a reliable algorithm here would be of much use for later stages. Other applications of beat tracking include use in musical information retrieval [2] and applications such as sychronisation to music (e.g. disco lights, automatic accompaniment).

Previous approaches fall into several distinct categories[1]: oscillating filters as typified by Scheirer [4] are one; autocorrelative methods, e.g. Foote [5] are another. Multiple hypothesis approaches such as Dixon [6] or Goto [7] are close to fully probabilistic approaches (e.g. Laroche [8] or Raphael [9]) in that they all evaluate the likelihood of a hypothesis set and choose the best; only the framework varies.

2. PROBLEM STATEMENT

Musical meter has been broken down into three levels [10]: the *pulse* which is the preferred human tapping tempo, the *tatum* which is the shortest regularly occurring interval and the *bar* or *measure* which is related to the rate of harmonic change and is essentially the bar line in notated music. This study centres on the estimation of the first of these, the pulse or beat and aims to estimate both the frequency and phase of this from musical audio signals.

3. MODEL AND METHOD

In summary, the approach taken here is to firstly extract onsets from the audio signal and to then use a particle filtering algorithm to track these through time and perform sequential estimation of the most likely beat.

3.1. Onset Detection

Reliable and accurate musical change detection is crucial to the success of any later algorithm. The onset detection falls into two categories; firstly there is detection of transient events which have strong energy changes associated with them, epitomised by drum sounds and secondly, there is detection of harmonic changes without large associated energy changes. For the first of these, the method approximately follows many algorithms in the literature: frequency bands, f, are separated and an energy evolution envelope $E_f(k)$ formed. A three point linear regression is used to find the gradient of $E_f(k)$ and peaks in this are detected. Low energy onsets are discarded and when there are closely spaced pairs, the lower amplitude one is also discarded. Three frequency bands were used: 20-200Hz to capture low frequency information; 200Hz-15kHz which captures most of the harmonic spectral region; and 15-22kHz which, contrary to the opinion of Duxbury [11], contains very clear transient information, generally without interference from harmonic components. With 44.1kHz sampling rate, the lower two frequency bands used a frame length of 1024 samples while the highest frequency band utilised a 256 sample window for extra time resolution.

Harmonic change detection is a harder problem. Here, long STFT windows (4096 samples) with short hop rate (1/8 frame) were used and a modified Kullback-Lieber distance measure used to detect spectral change:

$$d(n) = \log_2 \left(\frac{|X_k(n)|}{|X_{k-1}(n)|} \right) \tag{1}$$

$$D_{MKL}(k) = \sum_{n=1, d(n)>0}^{N/2} d(n) \tag{2}$$

where $X_k(n)$ is the n_{th} bin of the k_{th} frame of the STFT. The modified measure is thus tailored to accentuate positive energy change. Actually, only the region 40Hz-5kHz was considered and to pick peaks, a local average of the function was formed and then the maximum picked between each of the crossings of the actual function and the average. In summary, there are four vectors of onset observations, three from energy change detectors and one from a harmonic change detector.

The first author is kindly supported by the George and Lillian Schiff Foundation.

[1] Seppänen [3] contains a comprehensive literature review.

3.2. Beat Model

The model used in this study is loosely based on that of Cemgil [12]. Given the series of onset observations generated as above, the problem is to find a tempo profile which links them together and to assign each observation to a quantised "score" location.

We cast the problem as a jump Markov linear system (JLMS) [13],

$$\mathbf{x}_k = \Phi_k(\gamma_k)\mathbf{x}_{k-1} + \xi_k \tag{3}$$

$$\mathbf{y}_k = H_k\mathbf{x}_k + \nu_k \tag{4}$$

where $\mathbf{x}_k$ is the tempo process at iteration k and can be described as $\mathbf{x}_k = (\tau_k, \Delta_k)^t$. τ_k is then the estimated time of the k^{th} observation and Δ_k the tempo period, i.e. $\Delta_k = 60/T_k$ where T_k is the tempo in beats per minute. This is essentially a constant velocity process and the state innovation, ξ_k is modelled as zero mean Gaussian with covariance Q.

The second half of the problem is the association of observations to score locations, by which we mean the location on a traditionally notated score a human would use to represent the quantised position of the onset. Cemgil breaks a single beat into subdivisions of two and uses a prior related to the number of significant digits in the binary expansion of the quantised location. This was designed for midi signals and is not rich enough for the onsets generated from an audio signal which can be erroneous in their location or completely spurious. Thus we break the beat down into 24 locations, $c_k = \{1/24, 2/24, ...\}$ and assign a prior, $P(c_k) = exp(-log_2(d(c_k)))$ where $d(c_k)$ is the denominator of the fraction of c_k when expressed in its most reduced form; i.e. $d(3/24) = 8$, $d(36/24) = 2$ etc. The final ingredient of equation 3 is the state update matrix, $\Phi_k(\gamma_k)$ which is

$$\Phi_k(\gamma_k) = \begin{bmatrix} 1 & \gamma_k \\ 0 & 1 \end{bmatrix} \tag{5}$$

$$\gamma_k = c_k - c_{k-1} \tag{6}$$

This leads to the optimal state innovation covariance matrix

$$Q = q\begin{bmatrix} \frac{\gamma_k^3}{3} & \frac{\gamma_k^2}{2} \\ \frac{\gamma_k^2}{2} & \gamma_k \end{bmatrix} \tag{7}$$

for a constant velocity process.

Finally, there is the observation model. Given that there are four vectors of observations generated by the onset detectors, some of the onsets from different vectors will be assigned to the same score location. Therefore, a preprocessing step is applied to group the onsets from different vectors. A suitable measure is whether the onsets from the different streams fall within 50ms for transient onsets and with an additional tolerance of 30ms for harmonic onsets (which have lower time resolution). Inspection of the resulting grouped onsets shows that the inter-group separation is usually significantly less than the within-group time differences. Now, H_k becomes a function of the length j of the observation vector, $\mathbf{y}_k$ but is essentially j rows of the form $H_k = \begin{bmatrix} 1 & 0 \end{bmatrix}$. The observation error ν_k is also of length j and is modelled as zero mean Gaussian with covariance R where R is diagonal and the elements R_{jj} are related to whichever observation vector is being considered at $\mathbf{y}_k(j)$. Thus, conditional upon the c_k process, everything is modelled as linear Gaussian and the traditional Kalman filter [14] could be used.

3.2.1. Amplitude modelling

To this point, the algorithm will track tempo but not phase explicitly. So, we turn to the amplitude of the onsets and model the fact that we expect onsets in some locations in the bar to have higher energy than others. This can also be represented as a JMLS conditional upon c_k. The state equations are given by

$$\alpha_p^l = \Theta_p^l \alpha_{p-1}^l + \epsilon_p \tag{8}$$

$$a_p^l = D\alpha_p^l + \varsigma_p \tag{9}$$

where a_p^l is the amplitude of the p^{th} onset from the l^{th} observation vector and α_p^l is the state variable representing smoothed amplitude. An individual process is maintained for each observation function and updated only when a new observation from that level is encountered[2]. Θ_p is conditional upon c_p and c_{p-1}. In practice, after examining real data, an expected amplitude profile for each of the 24 subbeat locations was constructed and a 24x24 matrix built which contains the expected amplitude evolution from one location to another. This is then indexed by the currently considered and previously estimated location to find Θ_p. This is a weak assumption but it is expected that loud onsets will fall at metrically strong positions a majority of the time. For example, updating from a quaver to a crotchet location, the amplitude can be modelled as being likely to increase. The state and observation innovations, ϵ_p and ς_p are assumed to be zero mean Gaussian with large variances reflecting the low reliability of this process.

From now on, to avoid complicating the notation, the amplitude process will be represented without sums or products over the l conditioning. For each iteration, one or more of the amplitude processes will be updated. All non-updated processes are given a probability of 1 at that iteration and therefore the product of probabilities is kept consistent.

3.3. Methodology

Given the above system, a particle filtering algorithm can be used to estimate the posterior at any given iteration. The posterior which we wish to estimate is given by $p(c_{1:k}, \mathbf{x}_{1:k}, \alpha_{1:p}|\mathbf{y}_{1:k}, a_{1:p})$ but rather than work directly with this, one can perform so-called Rao-Blackwellisation [15] which has been proven to improve performance. The method breaks down the posterior into separate terms

$$\begin{aligned} p(c_{1:k}, \mathbf{x}_{1:k}, \alpha_{1:p}|\mathbf{y}_{1:k}, a_{1:p}) = p(\mathbf{x}_{1:k}|c_{1:k}, \mathbf{y}_{1:k}) \times \\ p(\alpha_{1:p}|c_{1:k}, a_{1:p})p(c_{1:k}|\mathbf{y}_{1:k}, a_{1:p}) \end{aligned} \tag{10}$$

where the terms $p(\mathbf{x}_{1:k}|c_{1:k}, \mathbf{y}_{1:k})$ and $p(\alpha_{1:p}|c_{1:k}, a_{1:p})$ can be deduced exactly by use of the traditional Kalman filter equations. Thus the only space to search over and perform recursion upon is that defined by $p(c_{1:k}|\mathbf{y}_{1:k}, a_{1:p})$ which is discrete but of impractically large size to search all possible paths.

Particle filtering represents the posterior as a set of weighted point estimates

$$p(c_{1:k}|\mathbf{y}_{1:k}, a_{1:p}) \simeq \sum_{i=1}^{N} w_{1:k}^{(i)} \delta_{\left(\mathbf{c}_{1:k}^{(i)}\right)} \left(d\mathbf{c}_{1:k}^{(i)}\right) \tag{11}$$

$$w_{1:k}^{(i)} = \frac{p(c_{1:k}|\mathbf{y}_{1:k}, a_{1:p})}{\pi(c_{1:k}|\mathbf{y}_{1:k}, a_{1:p})} \tag{12}$$

[2] Hence the conditioning on p rather than k; p then represents all the indexes up to iteration k where an observation from vector l is found.

where N is large and $\sum_{i=1}^{N} w_{1:k}^{(i)} = 1$. For more details on the theory, see [16]. The expression $\pi(c_{1:k}|\mathbf{y}_{1:k}, a_{1:p})$ is termed the importance distribution and the idea is to update each particle at the current iteration with a state drawn from this. Choice of importance distribution is critical to the performance of a particle filter algorithm.

By assuming that the distribution of c_k is dependent only upon $c_{1:k-1}$, $\mathbf{y}_{1:k}$ and $a_{1:p}$, the importance function can be factorised into terms such as $\pi(c_k|\mathbf{y}_{1:k}, a_{1:p}, c_{1:k-1})$. This allows recursion of the Rao-Blackwellised posterior

$$p(c_{1:k}|\mathbf{y}_{1:k}, a_{1:p}) \propto p(\mathbf{y}_k, a_p|\mathbf{y}_{1:k-1}, a_{1:p-1}, c_{1:k}) \times p(c_k|c_{k-1}) p(c_{1:k-1}|\mathbf{y}_{1:k-1}, a_{1:p-1}) \qquad (13)$$

where

$$p(\mathbf{y}_k, a_p|\mathbf{y}_{1:k-1}, a_{1:p-1}, c_{1:k}) = p(\mathbf{y}_k|\mathbf{y}_{1:k-1}, c_{1:k}) \times p(a_p|, a_{1:p-1}, c_{i:k}) \qquad (14)$$

and recursive updates to the weight are given by

$$w_k^{(i)} = w_{k-1}^{(i)} \times \frac{p(\mathbf{y}_k|\mathbf{y}_{1:k-1}, c_{1:k}^{(i)}) p(a_p|, a_{1:p-1}, c_{1:k}^{(i)}) p(c_k^{(i)}|c_{k-1}^{(i)})}{\pi(c_k^{(i)}|\mathbf{y}_{1:k}, a_{1:p}, c_{1:k-1}^{(i)})} \qquad (15)$$

The terms $p(\mathbf{y}_k|\mathbf{y}_{1:k-1}, c_{1:k})$ and $p(a_p|, a_{1:p-1}, c_{i:k})$ are calculated from the innovation (or residual) vector and covariance of the respective Kalman filters. $p(c_k|c_{k-1})$ is simplified to $p(c_k)$ and is hence the prior on score location as given in section 3.2.

3.3.1. Algorithm

The algorithm therefore proceeds as below. Given the set of potential states generated by propagating each particle, there are two ways of choosing a new set of updated particles: either stochastic selection or deterministic selection. The first proceeds in a similar manner to that described by Cemgil [12] where for each particle the new state is picked from the importance function with a given probability. Deterministic selection simply takes the best N particles from the whole set of propagated particles, thus giving $\pi(c_k|\mathbf{y}_{1:k}, a_{1:p}, c_{1:k-1}) = 1$. Both methods were tested and neither gave significantly better performance over the other for N of around 100 or greater.

Algorithm 1 Particle Filter Algorithm

```
• For $k = 1$
    – for $i = 1 : N$; draw $\mathbf{x}_1^{(i)}$, $a_1^{(i)}$ and $c_1^{(i)}$ from respective priors
• for $k = 2$ : end
    – for $i = 1 : N$; propagate particle $i$ to a set of new locations $c_k^{(s)}$ and evaluate weight update term for each of these. This generates $\pi(c_k|\mathbf{y}_{1:k}, a_{1:p}, c_{1:k-1}^{(i)})$.
    – for $i = 1 : N$; pick a new state for each particle from $\pi(c_k|\mathbf{y}_{1:k}, a_{1:p}, c_{1:k-1}^{(i)})$ and update weights according to Eqn. 15
```

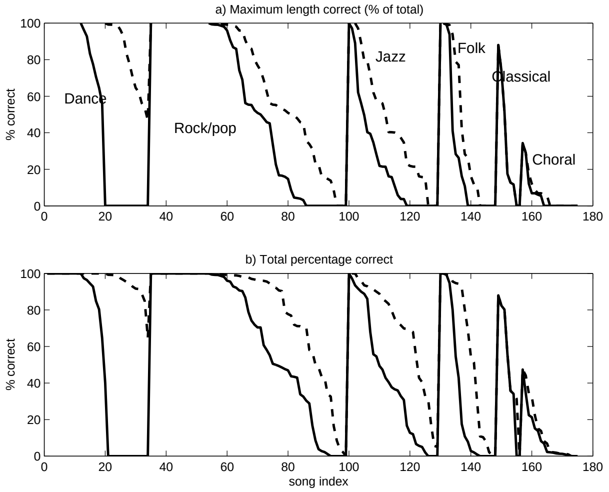

Figure 1: Results on test database. The solid line represents raw performance and the dashed line is performance after acceptable tracking errors have been taken into account.

3.3.2. Degeneracy

Particle filters suffer from degeneracy in that the posterior will eventually be represented by a single particle with high weight while many particles have negligible probability mass. Traditional PFs overcome this with resampling (see [16]) but both methods for particle update in the previous section implicitly include resampling. However, degeneracy still exists, in that the PF will tend to converge to a single c_k state, and a number of methods were explored for increasing the diversity of the particles. Firstly, jitter was added to the tempo process to increase local diversity. Secondly, a Metropolis-Hastings (MH) step [17] was used to explore jumps to alternative phases of the signal (i.e. to jump from tracking quavers to being on the beat). Also, an MH step to propose related tempos (i.e. doubling or halving the tracked tempo) was investigated but found to be counterproductive.

4. RESULTS

The algorithm was tested on a database of 175 examples from a wide range of styles, including dance rock/pop, jazz, folk, choral music and various examples of classical instrumental music. The average length of sample was about one minute and each was hand labelled for beat locations. The algorithm was deemed to be tracking the tempo correctly if there was a detected beat within 15% of the actual beat (measured as percentage of the interbeat period) and the tempo was correct to within 20% of the actual tempo. Two metrics are then proposed: firstly, the total percentage of beats correctly tracked and secondly, the longest consecutive region of correctly tracked beats expressed as a percentage of the total.

Given this, the raw results show on average 48.7% and 39.6%

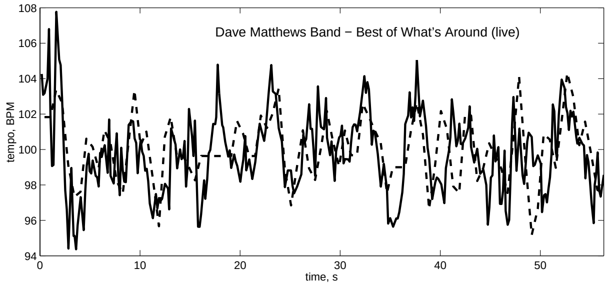

Figure 2: Tempo evolution for a correctly tracked example. Dashed line - hand labelled tempo; solid - estimated tempo.

success over the database. However, common tracking errors are a doubling or halving of tempo and tracking 180^o out of phase (i.e. tracking the off-beats) and after this is taken into account, performance is raised to 71.1% and 58.1% respectively. Finally, if choral music is excluded from the database (being hard for even a trained musician to follow and in the context here, very difficult to extract onsets from), performance is raised to 78.1% and 64.2%. A plot of the results is shown in figure 1 while figure 2 shows the tempo evolution of a correctly tracked example.

5. DISCUSSION

Overall, the algorithms presented above track the beat through music with comparable accuracy to the current state of the art: Klapuri [10] rates his algorithm at 57% success for longest sequence and 78% with total percentage correct when doubling and halving of tempo is allowed. Also for comparison, Scheirer's beat tracker [4] was run on the test database and performed at 22.7% and 35.7% respectively.

The main errors found in this study were also noted by Klapuri: tracking at double or half the tempo and phase inaccuracy by 50% where the tempo is still correctly tracked. The phase inaccuracy was a particular problem with dance music, though this is understandable on the basis that this style of music places much emphasis on the off-beat.

There are several crucial issues with the algorithm - firstly there is the reliable and consistent detection of onsets with high accuracy. If the onset detection is poor, there is no hope of any algorithm being able to track a tempo using these as features. This is an area for ongoing research and a future version of the algorithm may well include onset detection explicitly in the model. Secondly, there is the issue of particle diversity: there is a tendency for all the particles to converge to a single state and this lack of diversity can cause problems when regions of uncertainty are encountered. An MCMC move to introduce variation over the most recent iterations is possible but in order to be effective, it must propose for a whole set of iterations in a consistent manner and this is difficult to achieve; again, it is an area for future research. Finally, there are broader areas for investigation: a richer variety of tempo models should be considered, possibly as another JLMS acting as a hyperparameter selection device; measure level tracking should be introduced for extra robustness; and "swing", as in jazz music, should be explicitly modelled though the algorithm seems to be performing reasonably well without this. At this point, the algorithm will still be a generic beat tracking algorithm, and it is my contention that for further improvements, the algorithm will have to be tailored to the type of music it is attempting to beat track: the difference between the types of tempo evolutions and data expectations for a choral work and a dance tune are so large that any further improvements will have to be style specific.

6. REFERENCES

[1] A. Klapuri, "Automatic transcription of music," Master's thesis, Audio Research Group, University of Tampere, Finland, 1998.

[2] G. Tzanetakis and P. Cook, "Musical genre classification of audio signals," *IEEE Trans. Speech and Audio Processing*, vol. 10, no. 5, pp. 293–302, July 2002.

[3] J. Seppänen, "Tatum grid analysis of musical signals," in *IEEE Workshop on Applications of Signal Processing to Audio and Acoustics*, 2001, pp. 131–4.

[4] E. Scheirer, "Tempo and beat analysis of acoustical musical signals," *J. Acoust. Soc. Am.*, vol. 103, no. 1, pp. 588–601, Jan. 1998.

[5] J. Foote and S. Uchihashi, "The beat spectrum: a new approach to rhythm analysis," in *Proc. Int. Conf. on Multimedia and Expo (ICME)*, 2001.

[6] S. Dixon, "Automatic extraction of tempo and beat from expressive performaces," *J. NMR*, vol. 30, no. 1, pp. 39–58, 2001.

[7] M. Goto, "An audio-based real-time beat tracking system for music with or without drum-sounds," *J. of New Music Research*, vol. 30, no. 2, pp. 159–71, 2001.

[8] J. Laroche, "Estimating tempo, swing and beat locations in audio recordings," in *IEEE Workshop on Applications of Signal Processing to Audio and Acoustics*, 2001, pp. 135–8.

[9] C. Raphael, "Automated rhythm transcription," in *Proc. ISMIR*, 2001.

[10] Klapuri, "Musical meter estimation and music transcription," in *Proc. Cambridge Music Processing Colloquium*, 2003, pp. 40–45.

[11] C. Duxbury, M. Sandler, and M. Davies, "A hybrid approach to musical note detection," in *Proc. Digital Audio Effects Workshop (DAFx)*, Hamburg, 2002, pp. 33–8.

[12] A. Cemgil and B. Kappen, "Monte Carlo methods for tempo tracking and rhythm quantization," *J. Artifical Intelligence Research*, vol. 18, no. 45-81, 2003.

[13] A. Doucet, N. Gordon, and V. Krishnamurthy, "Particle filters for state estimation of jump Markov linear systems," *IEEE Trans. Signal Processing*, vol. 49, no. 3, pp. 613–24, Mar. 2001.

[14] S. Blackman and R. Popoli, *Design and Analysis of Modern Tracking Systems*. Artech House, 1999.

[15] G. Casella and C. Robert, "Rao-Blackwellisation of sampling schemes," *Biometrika*, vol. 83, no. 1, pp. 81–94, Mar. 1996.

[16] A. Doucet, "On sequential simulation-based methods for Bayesian filtering," CUED, Tech. Rep. CUED/F-INFENG/TR.310, 1998.

[17] W. Gilkes, S. Richardson, and D. Spiegelhalter, Eds., *Markov Chain Monte Carlo in Practice*. Chapman and Hall, 1996.