DECODING TEMPO AND TIMING VARIATIONS IN MUSIC RECORDINGS FROM BEAT ANNOTATIONS

Andrew Robertson
School of Electronic Engineering and Computer Science
andrew.robertson@eecs.qmul.ac.uk

ABSTRACT

This paper addresses the problem of determining tempo and timing data from a list of beat annotations. Whilst an approximation to the tempo can be calculated from the inter-beat interval, the annotations also include timing variations due to expressively timed events, phase shifts and errors in the annotation times. These deviations tend to propagate into the tempo graph and so tempo analysis methods tend to average over recent inter-beat intervals. However, whilst this minimises the effect such timing deviations have on the local tempo estimate, it also obscures the expressive timing devices used by the performer. Here we propose a more formal method for calculation of the optimal tempo path through use of an appropriate cost function that incorporates tempo change, phase shift and expressive timing.

1. INTRODUCTION

Musicologists are interested in how individual performers convey musical expression, which can manifest itself through control of dynamics, instrumental timbre and through tempo and timing variation. Honing [9] describes performed rhythm as consisting of three aspects: the rhythmic pattern, the tempo or speed of the performed pattern, and expressive timing deviations. Whilst the tempo can be understood as the rate at which beats occur, the onset time of a note is also dependent upon deviation from strict metrical time. Indeed, deviation from the score is a crucial aspect of musical performance and these variations have been found to be systematic [15]. Vercoe [16] characterises the relationship between score and performance "as if the musical score acts as a carrier signal for other things we prefer to process".

Gouyon and Dixon [8] present the difficulty of analysing performance data in that "the two dimensions of tempo and timing are projected onto the single axis of time". At the extreme, any tempo change can be represented as a sequence of timing changes and vice-versa. One simple way to represent tempo is to use the instantaneous inter-beat interval, but in doing so all expressive timing information has also been included. Desain and Honing [6] criticise the use of such "tempo curves" as meaningful representations of timing, arguing that expressive features, such as rubato, do not scale linearly with tempo, and that timing must be understood in relation to musical structure and phrasing. Whilst a simple moving average can help to smooth this estimate, these other timing deviations remain hidden within the tempo data and there is no explanation how these two aspects of timing might relate to each other.

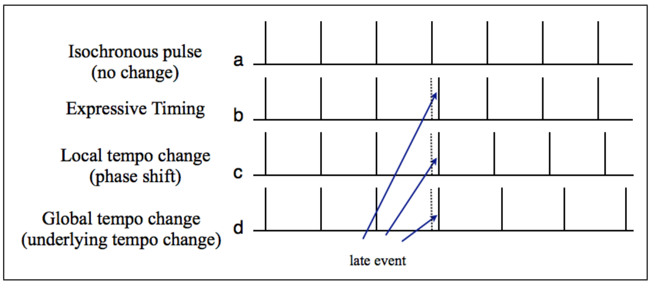

Figure 1. Four time lines illustrating the difference between the different timing variations (after Gouyon and Dixon [8]).

Despite these difficulties, it is still possible to attribute values to the tempo as it changes over throughout a piece, albeit with some inherent uncertainty. In rock and pop music, where the tempo is often approximately steady, beat trackers are successfully used to track tempo changes and when evaluated do so relatively well when compared with human listeners performing the same task [13]. The Performance Worm [7] provides real-time visualisation of tempo and dynamics by clustering inter-onset intervals. For scored music, Müller et al. [14] generate a tempo graph by aligning a "neutral" MIDI file, in which the tempo is constant, to the audio recording through the matching of chroma-onset features [14]. The tempo graph is calculated by using using windowing techniques to compute the average tempo in each local region.

1.1 Timing variations

Our model makes use of a framework provided by Gouyon and Dixon [8], who enumerate three types of timing variation: expressively timed events, local tempo variation or phase shift, and global tempo variation or tempo change. Figure 1 shows their illustration of each of these types for a single late event where the preceding events are a series of regularly spaced events. It should be noted that at the point in time where the event happens, it is unknown which type of timing variation has occurred.

In the case of expressively timed events, the event happens early or (in this case) late, but subsequent events are unaffected with respect to timing. The displacement occurs merely for the expressively timed event, but the underlying sequence is constantly spaced. In a local tempo change (or phase shift), there is a displacement for both the event and all subsequent events. So whilst the time between events, the underlying tempo, remains constant, the phase shift represents a variation in the interval. The global tempo change (or tempo variation) occurs when there is a change to the interval duration which continues in all subsequent intervals. This would be heard as a slowing down or speeding up of the events.

2. METHODOLOGY

Intuition might suggest that once the beat times in a recording are annotated that the instantaneous tempo is thereby known directly. We can calculate the tempo at annotation i at time t_i from the beat period $t_i - t_{i-1}$. If these annotation times are in milliseconds, the tempo in beats per minute (BPM) is $60000/(t_i - t_{i-1})$. However, in practice, such a tempo graph is often jagged and then requires smoothing to extract what is taken to be the underlying tempo. The reason for this is the conflation of tempo and timing (phase) variations, described above. Thus a local tempo change or phase shift will be represented by first a global tempo change in one direction and then a reverse change in the other. The smoothing process discards information about expressively timed events and phase shifts, as there is no explicit interpretation of the annotations in terms of potential timing variations.

Here, we propose a formal solution to this problem, which calculates the optimal timing variations according to a set of associated cost functions designed to penalise tempo change, expressive timing and phase shifts. By calculating the accumulated cost across a multitude of temporal locations (phases) and tempi, we can use the well-known dynamic programming technique to then trace the solution with least cost back through the song.

2.1 Input: Annotated beat times

For simplicity, we describe here how the method works for annotations at the beat level. However, the input can also be annotations at the note level, in which case the input contains both the event time and the quantised event location in beats and bars.

We shall assume that exact beat annotations exist for the audio recording. These take the form of a list of beat times, in seconds, and may have been generated either algorithmically or by hand. One program allowing the creation of annotated audio data is Sonic Visualiser [2] [1], designed to provide visualisation of audio analysis features using the VAMP plugin format. One such plugin is a beat tracking algorithm, based on work by Davies and Plumbley [5], which automatically labels the beats. It is also possible to manipulate these annotations, so that in cases where the beat is not exactly correct, it may be pushed earlier or later. Sonic Visualiser also supports the creation of hand annotated audio data by tapping the keyboard or use of a MIDI interface. This data may then be exported as a text file.

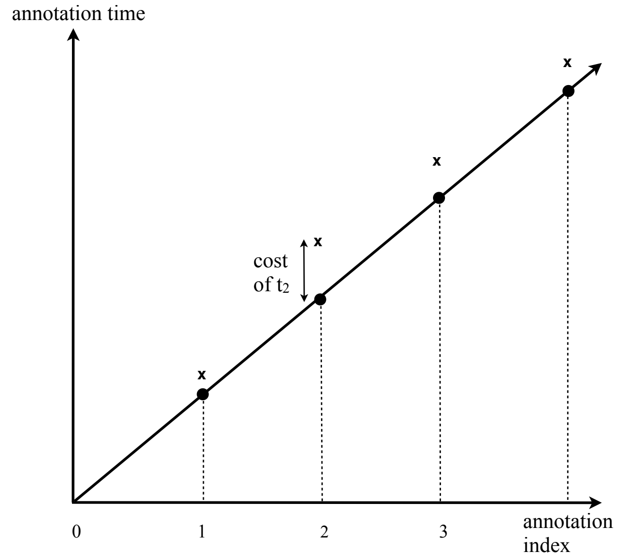

Figure 2. Illustration of the costs incurred by a sequence of isochronous pulses (circles) relative to the sequences of annotated beats $\{x_1, x_2, ...\}$. The cost for the pulse at annotation time t_2 is illustrated as the distance between the pulse and the annotation time.

The analysis proceeds on the assumption that the annotations indicate where the event actually occurred. The act of tapping along by hand, and use of the algorithmic beat tracker described above, will actually smooth the data by placing the beat towards the general trend rather than where each note onset happens. Thus if one wants to extract information about precise timing variations via this method, it would be advisable to then edit by hand so that the beat annotations are as close as possible to where the onsets occur in the audio. In the literature, a corresponding difference can also be found between predictive beat trackers which place beats causally and therefore smooth the output, and descriptive ones which place the beat after analysing the whole file and provide the ground truth of where the beat occurred [8].

2.2 Timing Transition costs

Given a set of annotated beat times as input, $\{t_0, t_1, t_2, ..\}$, we evaluate a total cost for each possible path of tempo and timing variations.

Let us define a beat path as consisting of a sequence of event times $\{\theta_0, \theta_1, \theta_2, ...\}$, each with an associated beat period of τ_i ms. These event times define the underlying beat and involve transitions in tempo and phase which incur costs. We can express each point on the path as a pair consisting of a phase location (the time of the event) and a tempo (as a beat period). Thus the point on the path corresponding to annotated time t_i is (θ_i, τ_i). Now we shall define the cost for this path and for the possible timing variations within it.

Firstly, the annotated beat time t_i might be expressively timed relative to this beat path, and the cost incurred is $|\theta_i - t_i|$, equivalent to the time difference in ms. In Figure 2, we see the cost of an annotated path relative to a series of isochronous pulses. The cost is simply the sum of the error between the two.

Secondly, the path may involve a phase shift or local tempo change. A tempo and phase pair $(\theta_{i-1}, \tau_{i-1})$ at annotation time t_{i-1} naturally implies that the next point on the path at annotation time t_i will be at $(\theta_{i-1}+\tau_{i-1}, \tau_{i-1})$. However, there may occur a phase shift of x ms, so that the next phase θ_i is in fact $\theta_{i-1}+\tau_{i-1}+x$. Then an additional cost is incurred of αx, where α is a parameter set by hand.

Thirdly, the path may involve a tempo change. Suppose we change from tempo of period τ_{i-1} to a tempo of $\tau_{i-1} + x$, making this the next tempo τ_i, we incur a cost of βx. The predicted point for such a transition would also have a phase location of $\theta_{i-1} + \tau_{i-1} + x$, although in this case due to the change in tempo.

To reflect the fact that we wish to penalise phase shifts and tempo changes, we set α and β by hand to values greater than 1. We have chosen α to be 1.4, and β to be 1.8 in practice although there is no definitive 'correct' value.

2.3 Updating the cost matrix

Let us define the cost matrix Γ_i to be all possible pairs of tempo and phase values, each with an associated cost. For each point (θ, τ) in Γ_i, we must consider all the possible transitions from points in Γ_{i-1}.

Supposing there was no change in tempo or phase, then a point (θ, τ) in Γ_{i-1} naturally suggests the next beat location at time t_i to be $\theta + \tau$ with the beat period remaining τ ms. We will employ dynamic programming to choose the minimum cost so far incurred on a path to (θ, τ) in Γ_i. This is done by working out the respective costs for all phase shifts and tempo changes to our new point and then choosing the minimum.

Observe that the point (θ, τ) in Γ_i can be reached from $(\theta - x - y, \tau - y)$ in Γ_{i-1} by a tempo transition of y ms (from $\tau - y$ to τ) and a subsequent phase shift of x ms, from the predicted event time of $\theta - x$ to θ. These incur costs of βy and αx respectively. We also need to compute the additional cost for the point, which is given by $|\theta - t_i|$, the discrepancy between the location of the beat and the annotation time. Then our full update equation is

$$\Gamma_i(\theta,\tau) = min_{x,y}\{\Gamma_{i-1}(\theta-x-y,\tau-y)+\alpha x+\beta y\}+|\theta-t_i|. \tag{1}$$

2.4 Backwards Path calculation

Having calculated the cost matrix Γ_i for each annotated beat times, t_i, we find the minimum point in the final matrix and the corresponding backwards path. Thus, we choose

$$(\theta_N, \tau_N) = min_{\theta,\tau}\Gamma_N(\theta,\tau) \tag{2}$$

Then we iterate back to find each previous point in the matrix that was chosen by Equation 1. This gives the complete path through the annotated beat times with the lowest cost for our parameters α and β. This path can be seen as the optimal explanation of the sequence of annotated beat times as a combination of tempo changes, phase shifts and expressively timed events.

2.5 Computational considerations

For our tempo analysis to be reasonably quick, we made use of some simplifications to reduce the computation time. By considering only those phase locations within occur within a fixed range either side of the beat annotation, we can discard points in the cost matrix which would almost certainly never occur. Similarly the tempo range was determined to be a fixed range either side of the interval between the two most recent annotations. These two ranges can be set by hand, depending on the nature of the piece.

Also, our data has a fixed temporal resolution. By choosing integers to represent note onset times, we have thereby chosen to use a precision of 1 ms for the resolution of both phase and beat period. However, by changing this to 2 ms or higher, the computation time can be reduced to a few seconds for a whole song without any significant degradation of the output.

3. PERFORMANCE ANALYSIS

The resulting tempo path is significantly more helpful when seeking to understand the global tempo changes in a performance than simply plotting the inter-beat intervals. We visualise the data using a standard tempo curve which plots the graph of tempo, or beat period, against the beat annotation index, i.e. plotting τ_i. Expressive timing information can be shown by placing a dot above or below this point (i, τ_i), whereby if the beat annotation occurs x ms after the location of the path point, then the dot is x ms above the tempo curve. In the figures below, for simplicity of presentation we have translated the beat period into the more commonly found representation as BPM. Whilst this thereby omits specific units for the expressive timing and phase shift information, we consider the benefits in understanding the tempo information to make this worthwhile.

3.1 The Beatles Dataset

An example of this can be seen in Figure 3. The input data was used was ground-truth annotations to the Beatles' song 'Taxman' from the album 'Revolver' [12]. The annotations were created in a semi-automatic manner, via the use of a beat tracking algorithm and then corrected by hand [4]. The fact that an algorithm was used does mean that some smoothing has taken place, however, our proposed decoding process still provides timing data that offers insightful information for musicologists.

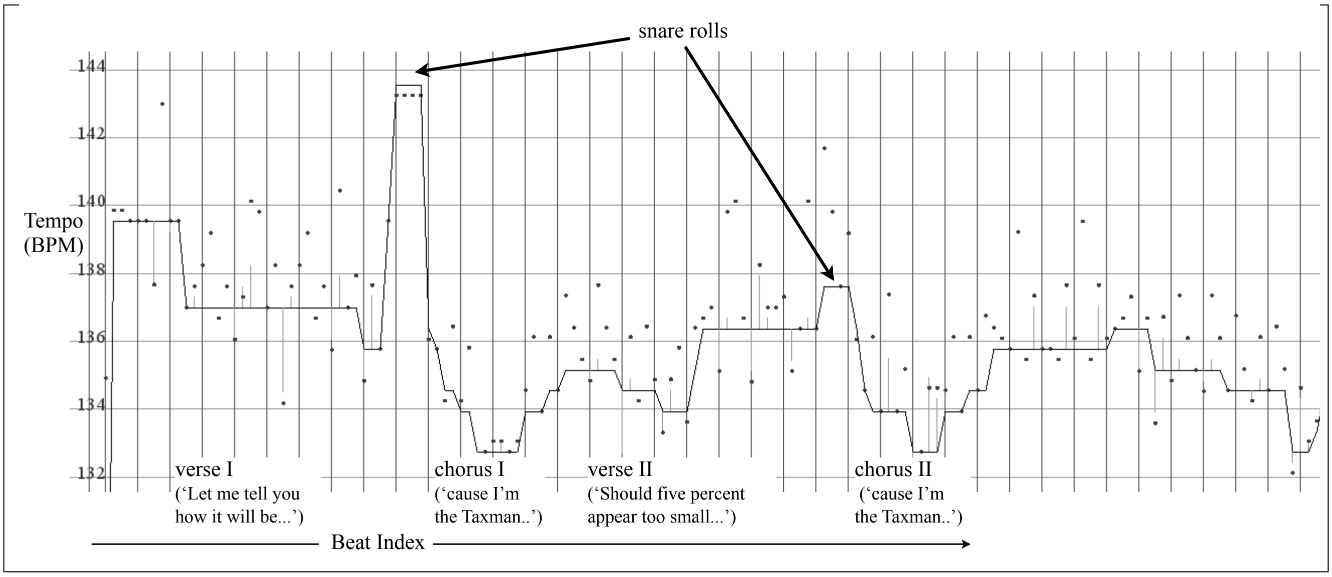

Figure 3. Tempo graph for The Beatles' 'Taxman'.

This analysis of the song indicates considerable complexity in Ringo Starr's time-keeping. He is both sensitive and in control of small fluctuations in tempo that generate a 'feel' to the different sections of the song. The decoded timing data displays clear small rises in tempo during the snare rolls that precede both the first and second choruses. There are then clear drops in tempo for both choruses of approximately 2 BPM, and this remains the case for later choruses beyond the scope of the Figure.

One can also observe a general trend in the expressive timing such that the timing of the second and fourth beats of the bar appears to be marginally later than the '1' and the '3'. On these beats, the song tends to feature the snare backbeat, as is common in rock music [10], and a regular guitar motif consisting of a staccato chord. Calculating the mean over the whole song confirms this, with the mean offsets being 0.25, 3.86, 0.82 and 2.18 ms for the respective beats. Drummers consider that placing the snare hit on '2' and '4' fractionally later, results in a more relaxed feel [11]. Such analysis supports the hypothesis that trends in microtime deviation lend a particular 'feel' to a song.

3.2 Beethoven's Moonlight Sonata

Chew [3] presents a detailed analysis of the timing variations in three different performances of Beethoven's Piano Sonata No. 14 in C Sharp Minor, Op. 27 No. 2: I. Adagio sostenuto, known as the 'Moonlight Sonata'. These recordings, by Daniel Barenboim[1], Artur Schnabel[2] and Maurizio Pollini[3], were initially used in an invited lecture by Jeanne Bamberger. This piece consists of repeated groups of four triplets in the right hand, with a movement between different chords. Such a repetitive structure suits it for revealing the tendencies of different performers with respect to their tempo and microtime variations. We have used of the same hand-annotated data, created using Sonic Visualiser.

In creating the tempo graphs, Chew comments on the necessity of smoothing to make the data understandable by the human eye, whilst warning that over-smoothing can result in important details being obscured. By use of the proposed method, we obtain smooth tempo graphs, but also preserve information about expressive timing and phase shifts.

The tempo graph for Pollini's performance is shown in Figure 4. Chew notes how the local minima of the tempo graph all occur on the bar boundaries. Bamberger contrasts this with the rendition by Schnabel, explaining that whereas other performers appear to 'stop' with each bass note at the beginning of the bar, Schnabel progresses through until the end of the first complete phrase after four bars "as if in one long breath". The extracted tempo and timing information for Schnabel's performance can be seen in Figure 5. Later in the piece, we can recognise a similar pattern to that exhibited by Pollini, whereby there is a slowing at the beginning of each bar.

This example can also serve to demonstrate some advantages of our proposed decoding method. The resulting tempo graph has less of the jagged edges that are still found in Chew's smoothed tempo graph. This is due to the projection of other timing data, expressive timing and phase shifts, onto the tempo curve. Instead, these quantities are made explicit and removed from the tempo curve, and thereby allowing us to calculate data relating to the phrasing of the notes. In this piece, we can observe that the third triplet eighth note exhibits a tendency to be marginally earlier than the first two notes of the bar. This would indicate that it thus begins earlier and is held fractionally longer. We have calculated the average deviation for each note and these results are presented in Table 1.

[1] On Beethoven: Moonlight, Pathtique and Appassionata Sonata CD Hamburg, Germany: Deutsche Grammophon GmbH.

[2] On Artur Schnabel CD United Kingdom: EMI Records Ltd.

[3] On Beethoven Piano Sonatas: Moonlight and Pastorale CD, Hamburg, Germany: Deutsche Grammophon GmbH.

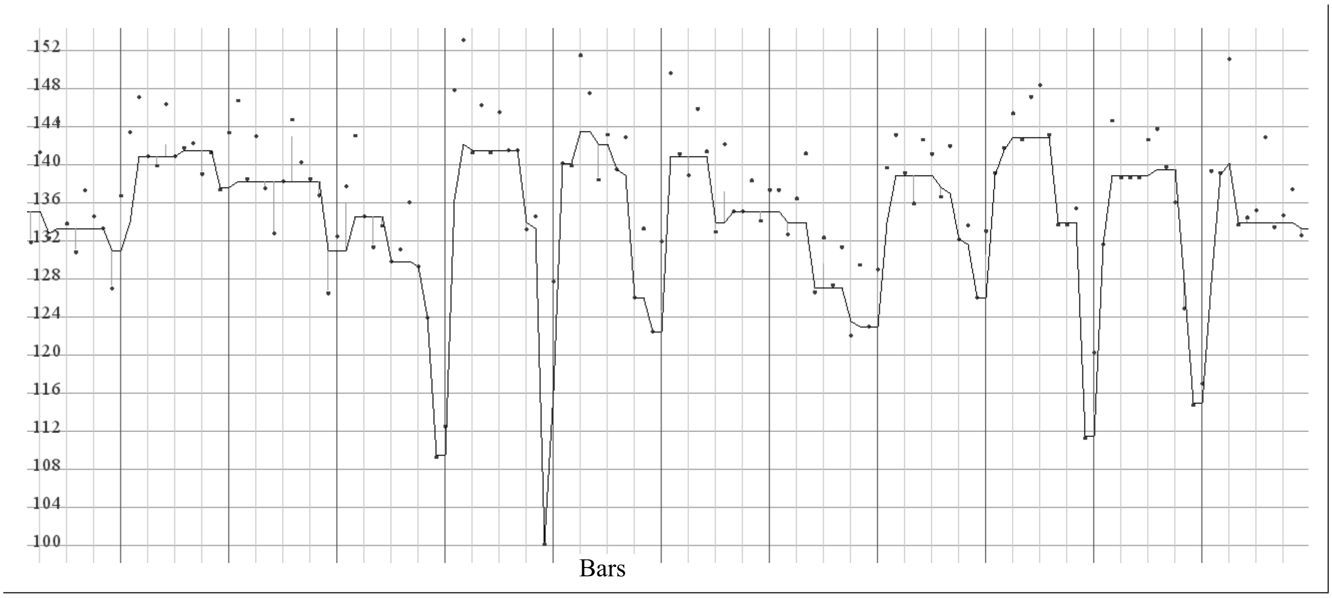

Figure 4. Tempo graph for Pollini's recording of Beethoven's Moonlight Sonata. The lighter vertical lines indicate crotchet boundaries and the darker vertical lines indicate bar boundaries. The expressive timing information is represented by the dots and phase shifts are represented by lines. Where the dot is above the tempo graph, the event is late relative to the time predicted by the underlying tempo; where the dot is below the event is early. Similarly a phase shift later is represented by a vertical line upwards from the tempo graph and a phase shift early by a line below the tempo graph. The tempo is indicated by BPM values to the left in 8 BPM intervals. The expressive timing and phase shift quantities are such that the

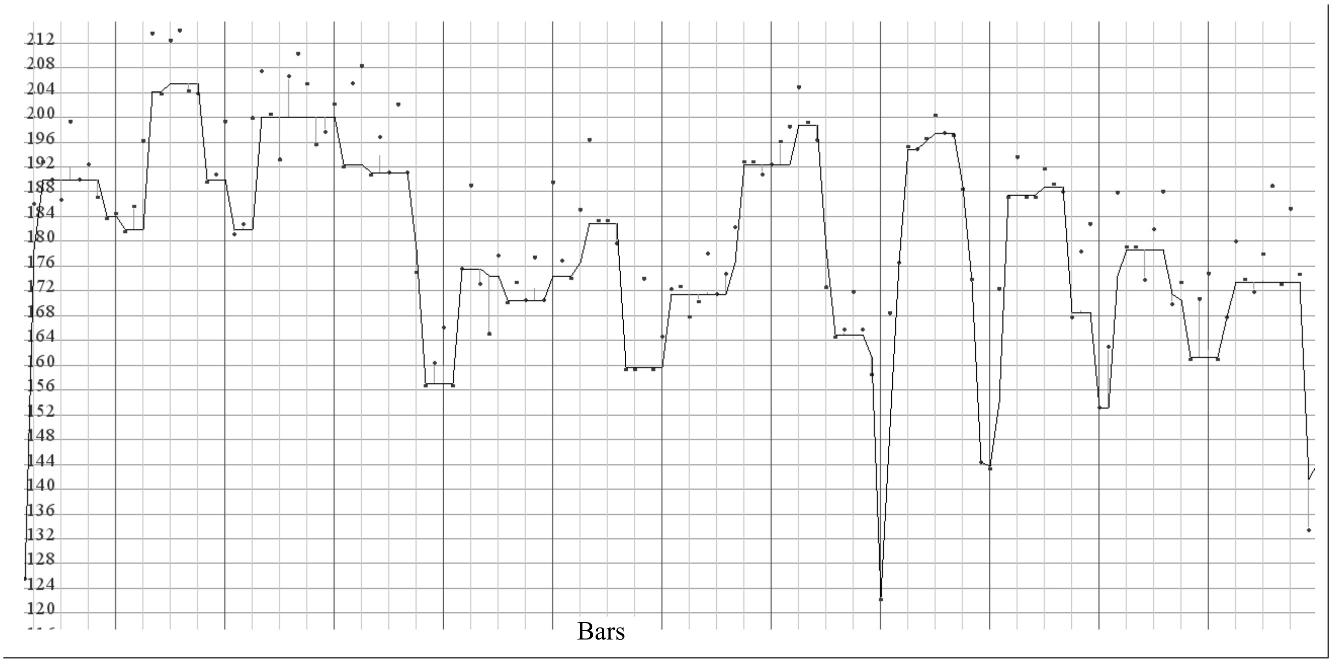

Figure 5. Tempo graph for Schnabel's recording of Beethoven's Moonlight Sonata. Again the lighter vertical lines indicate crotchet boundaries and the darker vertical lines indicate bar boundaries. the end of the first phrase is after four bars.

4. IMPLEMENTATION

The program was written in C++, using openFrameworks to provide visualisation using openGL libraries. The code is freely available for download at the Sound Software website [4], thereby allowing other researchers to import annotations. Both the resulting timing information and a file of the processed beat location times can then be exported as text files. Sonic Visualiser supports the loading of the processed annotations, which can then be sonified. In informal

[4] https://code.soundsoftware.ac.uk/projects/performance-timing-analyser

	Triplet note index		
Performer	1	2	3
Barenboim	7.27	2.56	0.83
Pollini	6.48	5.11	2.50
Schnabel	5.20	4.45	0.76

Table 1. Average deviation by triplet eighth note position in ms for the three performances.

tests, our processing algorithm appeared to have smoothed the data well, elimating timing errors whilst preserving the timing variations we are interested in.

5. CONCLUSION

In this paper, we present a new method for extracting the optimal tempo and timing path from a list of onset annotations. The output contains both tempo and expressive timing information for the optimal path according to our cost parameters. Such information enables a detailed musicological analysis of how performance timing data relates to musical structure. We have investigated how such data might be used in a classical case with the study of three performances of Beethoven's Moonlight Sonata, and in the rock and pop case through studying the timing of songs by The Beatles.

In future, we seek to extend our application of this method to the analysis of other annotated audio and develop a better understanding of how musicians make use of tempo and timing variations in expressive performance. We envisage that such work might also lead to improvements in the expressivity of computer-generated parts.

6. ACKNOWLEDGEMENTS

Thanks to Elaine Chew for making available the annotations for the Beethoven piano sonata recordings. Thanks to the EPSRC and the Royal Academy of Engineering for supporting this research.

7. REFERENCES

[1] C. Cannam, C. Landone, and M. Sandler. Sonic Visualiser: An open source application for viewing, analysing, and annotating music audio files. In *Proceedings of the ACM Multimedia 2010 International Conference, Firenze, Italy, October 2010.*, pages 1467–1468, 2010.

[2] Chris Cannam, Chris Landone, Mark B. Sandler, and J.P. Bello. The Sonic Visualiser: A visualisation platform for semantic descriptors from musical signals. In *Proceedings of the 7th International Conference on Music Information Retrieval (ISMIR-06)*, 2006.

[3] Elaine Chew. About time: Strategies of performance revealed in graphs. *Visions of Research in Music Education*, 20, 2012.

[4] M. E. P. Davies, N. Degara, and M. D. Plumbley. Evaluation methods for musical audio beat tracking algorithms. technical report c4dm-tr-09-06. Technical report, Queen Mary University of London, Centre for Digital Music., 2009.

[5] M. E. P. Davies and M. D. Plumbley. Context-dependent beat tracking of musical audio. *IEEE Transactions on Audio, Speech and Language Processing*, 15(3):1009–1020, 2007.

[6] Peter Desain and Henkjan Honing. Tempo curves considered harmful: A critical review of the representation of timing in computer music. In *Proceedings of International Computer Music Conference*, pages 143–149, 1991.

[7] Simon Dixon, Werner Goebl, and Gerhard Widmer. The Performance Worm: Real time visualisation based on langner's represen- tation. In *Proceedings of the International Computer Music Conference*, 2002.

[8] Fabien Gouyon and Simon Dixon. A review of automatic rhythm description systems. *Computer Music Journal*, 29(1):34–54, 2005.

[9] Henkjan Honing. From time to time: The representation of timing and tempo. *Computer Music Journal*, 25(3):50–61, 2002.

[10] Tommy Igoe. In the Pocket. Essential Grooves. Part 2. Funk. *Modern Drummer*, July 2006.

[11] Vijay Iyer. *Microstructures of Feel, Macrostructures of Sound: Embodied Cognition in West African and African-American Musics.* PhD thesis, University of California, Berkeley, 1998.

[12] M. Mauch, C. Cannam, M. Davies, S. Dixon, C. Harte, S. Kolozali, D. Tidhar, and M. Sandler. Omras2 metadata project 2009. In *Late-breaking session at the 10th International Conference on Music Information Retrieval (ISMIR 2009)*, 2009.

[13] 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.

[14] Meinard Müller, Verena Konz, Andi Scharfstein, Sebastian Ewert, and Michael Clausen. Toward automated extraction of tempo parameters from expressive music recordings. In *Proceedings of the International Society for Music Information Retrieval Conference (ISMIR), Kobe, Japan.*, 2009.

[15] Bruno H. Repp. Patterns of expressive timing in performances of a beethoven minuet by nineteen famous pianists. *Psychology of Music*, 22:157–167, 1995.

[16] Barry Vercoe and Miller Puckette. Synthetic Rehearsal, training the Synthetic Performer. In *Proceedings of the International Computer Music Conference (ICMC 1985)*, pages 275–278, 1985.