CHARACTERIZING TEMPO CHANGE IN MUSICAL PERFORMANCES

Roger B. Dannenberg
Carnegie Mellon University
Computer Science

Sukrit Mohan
National Institute of Technology, Surathkal
Computer Science

ABSTRACT

Tempo change is an essential feature of live music, yet it is difficult to measure or describe because tempo change can exist at many different scales, from inter-beat-time jitter to long-term drift over several minutes. We introduce a piece-wise linear tempo model as a representation for tempo analysis. We focus on music where tempo is nominally steady, e.g. jazz and rock. Tapped beat data was collected for music recordings, and tempo was approximated as piece-wise linear functions. We compare the steadiness of tempo in recordings by accomplished, professional artists and in those by amateur artists, and show that professionals are steadier. This work offers new insights into the nature of tempo change based on actual measurements. In principle, improved models of tempo change can be used to improve beat tracking reliability and accuracy. In addition to technical applications, observations of music practice are interesting from a musicological perspective, and our techniques might be applied to a wide range of studies in performance practice. Finally, we present an optimal function approximation algorithm that that has broader applications to representation and analysis in many computer music applications.

1. INTRODUCTION

Our goal is to study the nature of tempo change in music where steady tempo is the norm. In principle, to measure tempo, one can simply label beat times and calculate beats per second. In reality, beats are perceptual and have no direct physical manifestation. One can look for acoustical features such as drum beats or note onsets, or one can tap along with music and record tap times, but either way, the observed times are the result of human gestures which are ultimately imprecise. The error in beat time estimates will give rise to errors in tempo estimation. Smoothing to improve long-term tempo estimates tends to remove variation in the shorter term, but it is the short term tempo variation that has the highest acceleration and is often the most interesting.

Therefore, we need to model tempo in a representation that largely ignores short-term variations due to jitter in beat times, yet captures the intricacies of tempo variations. Tempo transcribed by beat tracking is prone to outliers and random occurrences of incorrect beat labels. Our approach to model tempo changes segments a performance into regions in which tempo is changing approximately linearly. A segmented tempo curve has the advantage of capturing the regularity and predictability of tempo in the short term, while still allowing us to model significant tempo changes that occur occasionally. From this abstraction, we can perform a statistical analysis of tempo variations in real-time performances.

Musical tempo variation has been studied extensively, but most existing studies focus on Western art music and emphasize expressive and large tempo variation. Linear and polynomial models (also used here) of *accelerandi* and *ritardandi* are common. An article by Honing [4] presents a nice introduction and references to many other works. To our knowledge, this literature does not address the tempo variation in popular music. Our work is novel in that it introduces some optimal curve-fitting techniques to music analysis, and we study the typically small tempo variation in popular music.

Based on the data obtained from our tempo curve approximations, we perform a study comparing professional artists and amateur musicians. We study the range of tempo with respect to the mean, and the magnitude of tempo changes, and the degree of acceleration or deceleration (*accelerando* or *ritardando*).

One of the motivations for studying tempo change is to improve beat tracking systems, an important problem in Computer Music research. Performers rarely maintain a constant tempo throughout an entire performance, so beat trackers must adapt to tempo changes. Assuming that tempo is very steady tends to make a beat tracker more robust in the face of ambiguous input where beats are unclear. However, if the assumption is too restrictive, the beat tracker will fail to adapt to real tempo changes. Thus, the rate and extent to which tempo can change are important parameters in these systems, but there are few studies of actual performance practice to inform the design of beat trackers.

Interactive music systems can use simple interfaces such as foot tapping to acquire the tempo. Even in such a simple system, tempo models are useful. Linear regression can be used to overcome jitter in the previous foot taps and more accurately estimate the tempo and predict future beat times. However, the best regression length depends upon the expected amount of tempo variation. Thus, in order to understand how to predict the time of the next beat accurately, given previous beat time

estimates, first we must study the amount of tempo variation that actually occurs in live musical performances.

The remainder of this paper is organized as follows: Section 2 describes how accurate beat data has been obtained using two sets of manually annotated tap data. Section 3 presents the dynamic programming algorithm used for converting the noisy measured tempo curve to a segmented tempo curve approximation. In Section 4, we present the statistics showing differences in the tempo variation of professionals and amateurs. We describe our conclusions and plans for future work in Section 5.

2. OBTAINING BEAT TIMES

A common technique for beat annotation is to tap along with the audio and record the tap times. Labelling beats by hand is subject to human inaccuracy. On the other hand, automatic labeling is not reliable for most music, and even using acoustic features is subject to question because these features are, like tapping, the results of human gesture. Furthermore, audio peaks or other features may not align with perceptual beat times [8]. For this study, we accept tap-timing jitter as inevitable, and our goal is to study longer-term phenomena.

Tempo is measured by tapping along with recorded music. We tap on or near a microphone and record the taps using the same audio system that is playing the music. This ensures that the taps will synchronized with the music except for some audio playback and recording latency. The taps are converted to a sequence of times by scanning the audio for high amplitude tap onsets following silence. We tap twice and compare the two sequences of taps in order to detect and remove errors. We can also use the pairs of taps to characterize the tapping accuracy [2]. Taps collected in this manner typically have a Gaussian distribution with a standard deviation of about 30ms. By averaging the two tap times for each beat, the standard deviation is reduced to about 20ms.

If we plot beat number as a function of time, we get a beat-vs-time curve. We can estimate the tempo (beats per second) as the reciprocal of the period between each successive pair of beats. We call this the tempo-vs-time or simply "tempo" curve (see Figure 1).

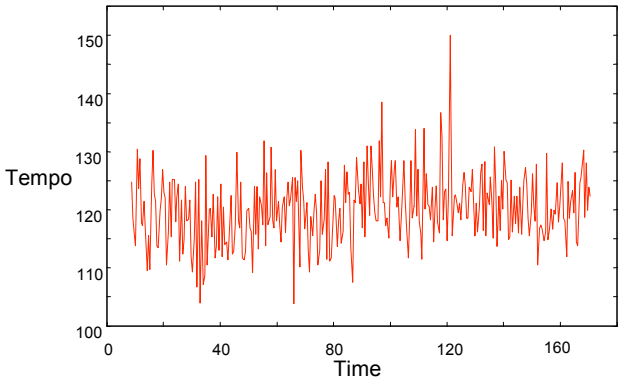

Figure 1. Tempo plot calculated from successive beat times

3. MODELLING TEMPO VARIATION

The tempo curve obtained by taps is noisy and implies that tempo fluctuates a lot at every beat. Our goal is to simplify this noisy tempo curve and convert it into a cleaner, more realistic representation of tempo. Tempo smoothing is one method of reducing the tempo graph to a more realistic representation. This raises the question: How much smoothing? An answer could be: Smooth on a time scale where tempo is essentially steady. But the original purpose of this study is to determine how tempo changes at different time scales. How can we do smoothing without assuming the answer to our question? We need an approach by which we can remove noise and model the "real" local tempo, but at the same time model significant tempo variations that occur in the performance.

In our approach, we segment a performance into regions in which tempo is changing approximately linearly. This is advantageous since it models a predictable tempo in the short term, but at the same time it can model significant sudden tempo changes when necessary to fit the data. Also, a piecewise tempo curve can be analyzed quantitatively to obtain important results regarding when, and at what rate tempo changes occur in musical performances.

It is tempting and common to model tempo as a function from beat number to time, but it is also common to use analogies to velocity and acceleration to reason about tempo and tempo change. Under this physical analogy, tempo is the analog of velocity, and beats are the analog of position. Thus, the beat-vs-time curve is a function from time to beat, not beat to time. The derivative of this curve is the instantaneous tempo. If we assume tempo change by constant acceleration, i.e. the tempo curve is piecewise linear, then the function from time to beat number is a second-order polynomial:

$$f(t) = at^2 + bt + c \qquad (1)$$

Let $\theta_{i,tap}$ be the tapped time of beat i, and $\theta_{i,approx}$ be the time of the beat generated by the approximated tempo. To approximate the entire sequence $\theta_{i,tap}$ of length n, we will use multiple polynomials. We choose a set of $m+1$ inflection point positions p_i, for $0 \le i < m$. We then find the best fitting polynomials from $\theta_{p0,tap}$ to $\theta_{p1,tap}$, $\theta_{p1,tap}$ to $\theta_{p2,tap}$, …, and $\theta_{pm-1,tap}$ to $\theta_{pm,tap}$. The polynomials are constrained to pass through the inflection points, so $\theta_{po,approx} = \theta_{p0,tap}$, $\theta_{p1,approx} = \theta_{p1,tap}$, etc.

Note that we have switched from discussing tempo-vs-time curves to their integrals – beat-vs-time curves – such as $f(t)$. We work with beat-to-time curves so that we can measure error in terms of predicted beat times. After curve fitting, we can always take the derivative to obtain the tempo curve.

Our intuition tells us to minimize *timing* errors when fitting polynomials to data, whereas the physics analogy and standard regression formulas would lead us to minimize errors in *position* or *beat*. Thus, we will often

work with the inverse function, $f^{-1}(\cdot)$, that maps beat numbers to time.

The sum-of-squares error we wish to minimize in fitting a curve between points *i* and *j* is:

$$\varepsilon_{i,j} = \sum_{i<k<j}\left(\theta_{k,tap} - \theta_{k,approx}\right)^2 \quad (2)$$

where $\theta_{k,approx} = f^{-1}(k)$, the inverse of *f* as defined above, which maps beat number *k* to time.

Minimizing errors to best to approximate the tapped data is a good start, but the approximation must also offer a plausible fit to the tapped data. We require that the approximated beat times, $\theta_{k,approx}$ fall within τ of the tapped times (see Figure 2):

$$\left|\theta_{i,tap} - \theta_{i,approx}\right| \leq \tau \quad (3)$$

Given this criterion, it might seem more natural to minimize the absolute error rather than the sum of squares, but a least squares approach seems to be better at capturing the notion of overall goodness of fit. Thus, we evaluate the goodness by summing the squared error terms over all the segments:

$$\varepsilon_{all} = \varepsilon_{p0,p1} + \varepsilon_{p1,p2} + \ldots + \varepsilon_{pm-1,pm} \quad (4)$$

Thus, the problem of modeling beat-vs-time curves is defined as such: *find the minimum number of second-order polynomial curves that form a piecewise approximation of the observed tap times to within an error threshold τ. Given this minimum number of curves, compute the specific curves that minimize the sum of the squared timing errors.* The tempo curve we will use for statistical analysis is just the derivative of the beat-vs-time curve.

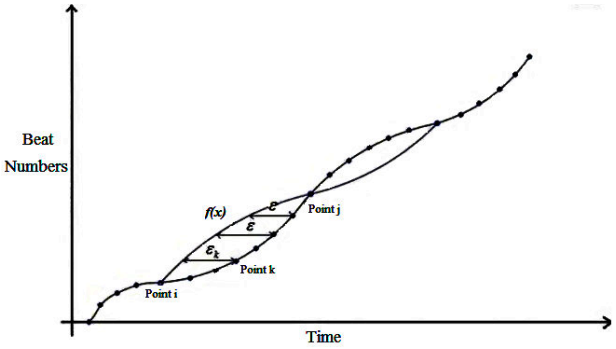

Figure 2. Quadratic curve fitting between true beat points such that error lies within threshold.

In Figure 2, each of the values of $\varepsilon_1, \varepsilon_2, \ldots, \varepsilon_k \leq \tau$.

3.1. Algorithm

This curve-fitting problem is difficult because there are many ways to divide the beat sequence into regions of constant acceleration. The number of ways to choose inflection points that divide the sequence of beats into segments is exponential in the number of beats, specifically $\binom{n}{m-1}$ where *n* is the number of points and *m* is the number of segments. Enumerating all possible choices and testing to find the optimal fit is not feasible. Instead, the algorithm uses dynamic programming to find the optimal solution, using a variation of methods described by Glus [3]. The "trick" is to store solutions to sub-problems in a table to avoid recomputing them. The matrix $L_{i,j}$ holds the total error for a single polynomial fit between points *i* and *j*. The matrix $D_{m,i}$ holds the optimal (smallest possible) error using up to *m* segments to fit the first *i* points in the total sequence. Each matrix can be computed in $O(n^3)$ time, making the algorithm $O(n^3)$: slow, but quite tractable. The computation of each matrix is now given in detail.

3.1.1. Creating the Look-up Table

An $n \times n$ look-up table L is created to hold data regarding the error values and the best-fitting quadratic curve between any two arbitrary points i and j on the beat-vs-time curve. Note that we want to minimize timing errors produced by a mapping from beat to time (our f^{-1}), but the standard least-squares fit of a polynomial (f) will minimize beat errors. Our solution begins by fitting a polynomial (f) to the data. Then, we compute the inverse f^{-1} of f in terms of coefficients *a*, *b*, and *c*. Finally, we use gradient descent to adjust *a*, *b*, and *c* (but since the curve is constrained to meet the first and last points in the sequence, there is really only one free parameter) to optimize the fit of f^{-1}. Since we are starting with an excellent approximation, this step converges rapidly, and for the purposes of algorithmic complexity, we count this step as O(*n*).

Once we have a minimum error curve between points i and j, we check to see whether the error at each point, ε_k lies within the threshold τ. If $\varepsilon_k \leq \tau$ then we add this error value to a sum $\varepsilon_{i,j}$, (initially 0). However, if for some point *k*, $\varepsilon_k > \tau$, then we set $\varepsilon_{i,j} = \infty$. The look-up table $L_{i,j}$ is set to the final $\varepsilon_{i,j}$ value. (We also save the coefficients of the best-fitting polynomial.) This procedure is carried out for every pair of points (i, j) such that $0 \leq i < j < n$.

3.1.2. Create the Error Table

The next step utilizes a dynamic programming approach to find the minimum error when choosing *m* inflection points out of a sequence of *n* points. The error-table D is also an $n \times n$ table. The value of $D_{m,i}$ is the total error (ε_{all}) given *m* inflection points and spanning the first *i* out of *n* beats. The algorithm computes each matrix value in terms of previously computed values until the last value is obtained.

Figure 3 shows how to compute the value of $D_{m,n-1}$ in terms of $D_{m-1,0}$ through $D_{m-1,n-2}$. Looking at the figure, it is obvious that if there are *m* inflection points (for *m*-1 segments), then the last point must be point *n*-1,

and the next to last point must be somewhere between m and n-2 (inclusive). If the next to last point is i, then there are m-1 inflection points from 0 to i, and $D_{m-1,i}$ tells us the sum of squared errors up to point i. $L_{i,n-1}$ gives us the error of the single remaining segment from i to n-1, so $D_{m-1,i} + L_{i,n-1}$ is the total error *if the next-to-last inflection point is at i*. Thus, the optimal (smallest) error is the minimum over i of $D_{m-1,i} + L_{i,n-1}$.

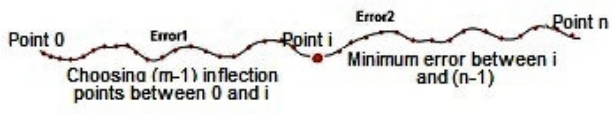

Figure 3. Understanding the error table dynamic programming algorithm

Note that we need to initialize the first row of D, but we have already computed the cases with only one segment in L, so we simply set $D_{1,i}$ to $L_{0,i}$ for each i.

Thus, by dynamic programming we obtain the final column of the error-table, which describes the total error when m inflection points are chosen between points 0 and $n-1$.

3.1.3. *Extract Number of Inflection Points*

Our goal is to find the minimal number of segments. We start to traverse from the bottom of the final column of D. It is likely that the first value is infinity (recall that we set the error value to infinity if *any* error exceeds the limit τ). This occurs because no single curve is likely to be a good fit to all the points. As we scan the column, eventually we will reach an error value less than infinity. The row number tells us the minimum number of inflection points needed to fit curves within the error limit τ. Once we know the number of segments, we can reconstruct what they are. For example, we can use an auxiliary matrix to save the value of i that minimized $D_{m-1,i} + L_{i,n-1}$ for each pair (m, i) and use this data to reconstruct the full set of m segments covering beats 0 to n-1. For further details, please contact the authors for an implementation in the Java programming language.

3.1.4. *Optimizations*

Rather than computing the entire $n \times n$ matrix D, we can compute one row at a time. When the last element of a row is not infinite, we can stop because we now have all the information we need to compute the best curve fit using the minimal number of segments.

When computing an element of L, if the error at (i, j) is infinite (meaning some timing error is greater than τ), then the polynomial approximation from i to j+1 is not likely to fit the data any better (although it is theoretically possible). Thus, when we observe an infinite error value for the curves from i to j, i to j+1, and so on until i to j+9, we simply stop fitting curves and fill in the rest of the row with infinity. Since it is unlikely that a single polynomial will fit a very long sequence of tap times, the true computation time for L, with this added heuristic, is closer to O(n^2) than O(n^3). Similarly, it is sometimes possible to determine an element of D is infinite without looping through all possible combinations of $D_{m-1,i} + L_{i,n-1}$. If one can place an upper bound on the length of any segment, the algorithm complexity becomes O(n^2).

Thus, using this algorithm, we have successfully generated a piecewise quadratic approximation to the beat-time curve with the minimum number of inflection points, such that the curve lies within the error threshold at the temporal positions of true beats.

Taking the derivative of the newly generated beat-time curve, we get our piecewise linear tempo graph which effectively models tempo in live performances.

3.1.5. *Value of* τ

We experimented with various different values for the error threshold τ, to see what value would best model the tempo curve accurately. We decided on using a threshold value of 60ms to model the approximate tempo graph, since this seemed to fit best with our perception of when sudden tempo changes occurred, both in terms of listening and of visually inspecting the tempo curve graphs. As can be seen in Figure 5, the computed curve gives a good summary of tempo trends in spite of the noisy tempo values implied by the beat-to-beat time intervals.

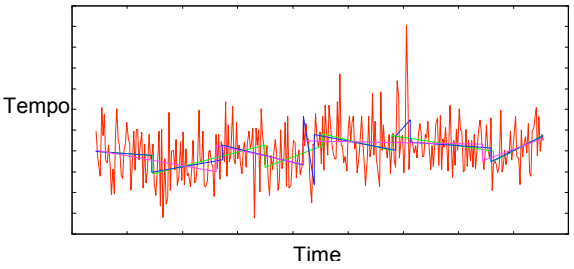

Figure 4. Approximation graphs for different thresholds

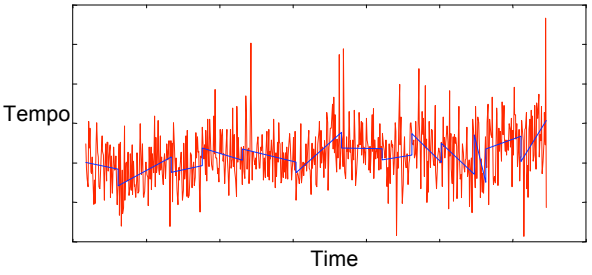

Figure 5. Approximation graph for threshold 60ms

4. STATISTICAL ANALYSIS

We collected tap data for a diverse collection of professional and amateur recordings. Our professional recordings collected comprised of Miles Davis' "Kind of Blue" album and the Beatles "Please Please Me" album.

We chose these particular albums for their relative resemblance to live recordings and for their difference in genres. "Kind of Blue" was recorded with minimal practice among the musicians playing. The entire album has been well documented in literature [6]. "Please Please Me" was also recorded in one continuous session without any modifications. These albums are ideal for studying live performance by professional musicians.

Our amateur recordings were collected over a span of a few months, recording practice sessions and live performances of amateur artists (including some paid performances). The amateur data set consists of amateur rock recordings and some jazz performances. Table 1 summarizes the data collected.

	Duration (s)	No of Beats	No of Inflection Pts
Ama Jazz	12739	16898	467
Pro Jazz	2274	4424	63
Ama Rock	2749	6827	156
Pro Rock	1783	3689	66

Table 1. Overall statistics of data collected

The features that we want to compare for professionals and amateurs are *i)* tempo relative to the median tempo for a song, *ii)* relative tempo acceleration. Both these features are related to the steadiness of tempo in live musical performances.

Tempo relative to the median tempo represents the actual tempo normalized by the median tempo within a piece. If tempo is always steady, this value will always equal one, thus we can compare pieces of differing tempo. The second feature, tempo acceleration (beats/second^2) represents the rate of change of tempo. One might expect acceleration to be higher with faster tempos than slower ones. Therefore, we normalize tempo acceleration by the tempo to obtain *relative acceleration.* Both tempo relative to the median (*relative tempo* in the graphs, Figure 6) and relative acceleration (Figure 7) are plotted as histograms to study their distributions.

We notice that amateur performers tend to deviate from the mean tempo a lot more than professionals. The histograms show they slow down to as much as approximately -20% from the mean tempo. However, during the course of one performance, they may speed up to +10% of the mean performance tempo. It is also observed that amateur rock performers have a roughly similar distribution. Professional performers are much steadier in their performances. From the data obtained, we note that the tempo for professional rock and jazz performers stays within one or two percent of the mean tempo. For amateur rock performers, deviation ranges between about –10% and +6%, whereas for amateur jazz performers it ranges between ±5%.

Since the mean relative tempo is expected to be 1 and the mean relative acceleration is expected to be 0 (unless there is a general trend to speed up or slow down), we can characterize the tempo steadiness by the variance of these values.

From the histograms, it seems clear that the variance is higher in amateur performances than in professional performances. We used Levene's test for homogeneity of variance [7] to compare professional vs. amateur variance of relative tempo in rock, relative tempo in jazz, relative acceleration in rock, and relative acceleration in jazz. In all cases, variance of the professional data is lower than that of the corresponding amateur data ($p < 0.01$).

5. CONCLUSION

The aim of this research is to understand how tempo changes in live musical performances. We developed a mathematical model by means of which noisy hand-tapped beat data can be converted into a compact summary which facilitates the study of tempo variations in live performances. In particular, the model assigns a rate of tempo change to every time point in the performance, allowing us to quantify the steadiness of tempo. We applied our model to data collected from amateur and professional "live" musical performances. We observed that the variance from the mean tempo of the performance is dependent on the artist's experience. Professional artists deviate from the mean tempo less than amateurs.

Our models were motivated by the problem of predicting the next beat time given beat times in the past. This is useful for computer accompaniment systems where accurate timing is important. We believe our characterization of the statistical distribution of tempo change could also be useful in beat tracking.

Our models could also be useful for creating more natural-sounding synthesized performances. Rock-steady tempo gives the impression of mechanical production. Some tempo variation according to statistical models based on real performance data could be a good way to generate musically plausible tempo variation.

Our algorithm to fit polynomials to tempo curves can be adapted to fit curves to any data. An apparently open problem in the computer music literature is finding the best piecewise approximation to amplitude and spectral envelopes. For example, Horner and Beauchamp [5] explored this problem and suggested genetic algorithms to search for good approximations. Our approach can be adapted to many curve-fitting and envelope approximation problems, producing optimal results with reasonable computation times, eliminating the need for heuristic search.

It should be mentioned that we used a fairly wide range of amateur data but collected professional data from only two groups. In the future, we hope to analyze a larger set of professional groups to rule out the possibility that the chosen works are atypical. We also plan to look for musical explanations for the tempo changes we observe in the data.

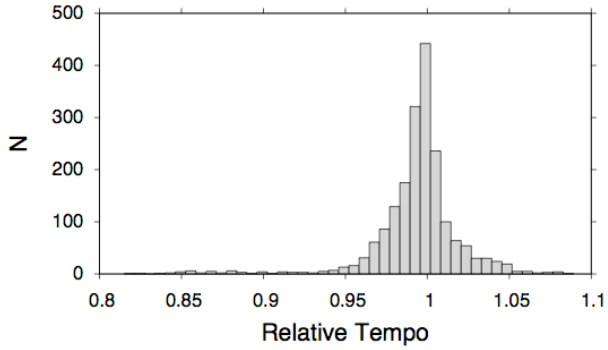

Figure 6(a). Histogram of relative tempo within continuously played sections – jazz amateur.

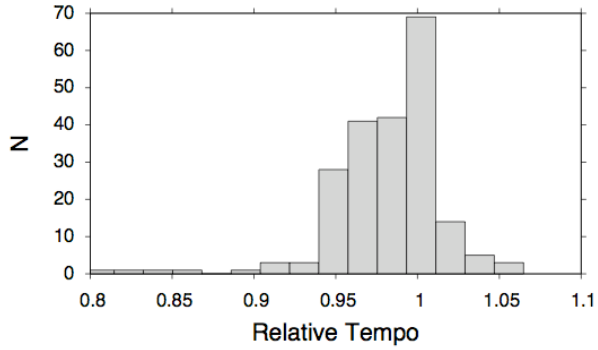

Figure 6(b). Histogram of relative tempo within continuously played sections – rock amateur.

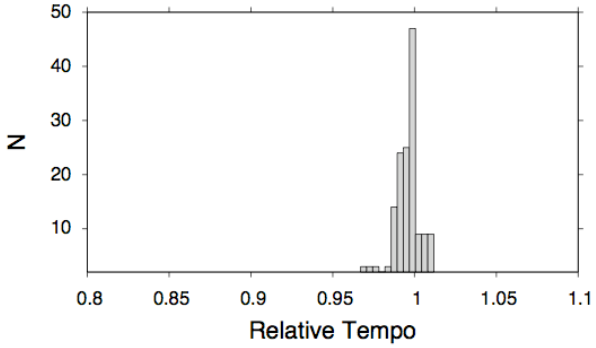

Figure 6(c). Histogram of relative tempo – jazz professiona (Miles Davis).

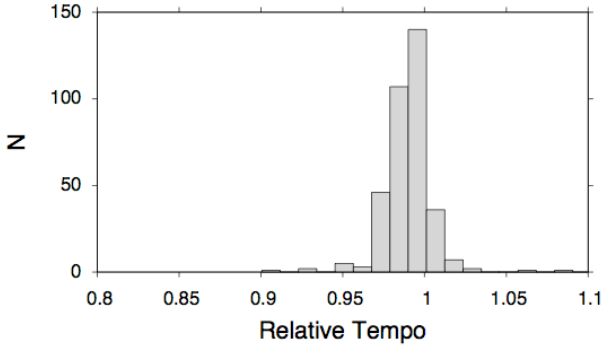

Figure 6(d). Histogram of relative tempo – rock professional (Beatles).

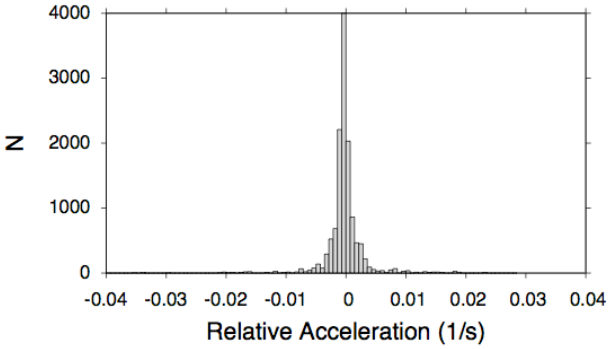

Figure 7(a). Histogram of relative acceleration within continuously played sections – jazz amateur.

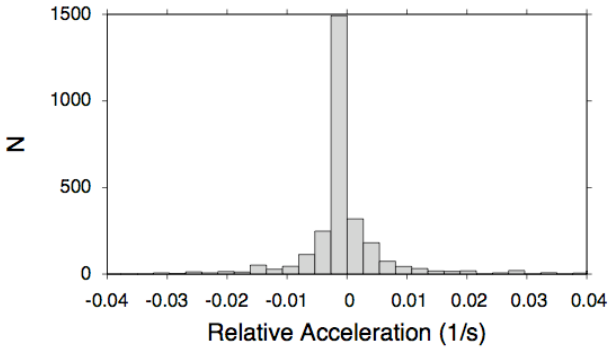

Figure 7(b). Histogram of relative acceleration within continuously played sections – rock amateur.

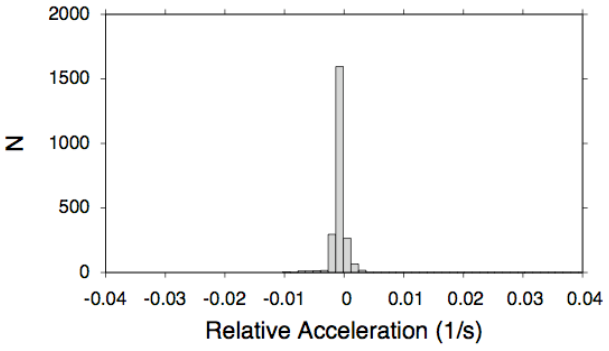

Figure 7(c). Histogram of relative acceleration – jazz professional (Miles Davis)

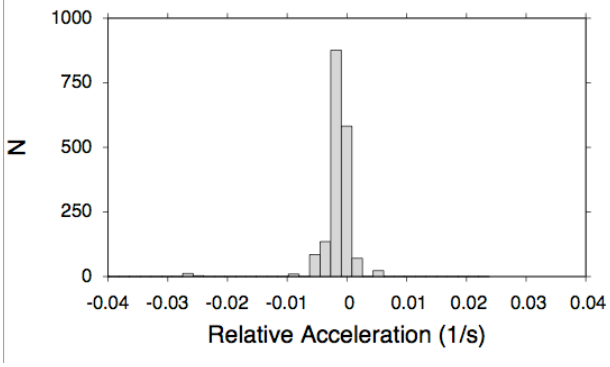

Figure 7(d). Histogram of relative acceleration – rock professional (Beatles).

6. ACKNOWLEDGEMENTS

We would like to thank Madhur Shukla for his assistance. This work was funded in part by grants from Microsoft Research through the Computational Thinking Center at Carnegie Mellon and the National Science Foundation Grant No. 0855958.

7. REFERENCES

[1] Dannenberg, R. "Towards Automated Holistic Beat Tracking, Music Analysis, and Understanding." *ISMIR 2005 6th International Conference on Music Information Retrieval Proceedings*, London: Queen Mary, University of London, (2005), pp. 366-373.

[2] Dannenberg, R. and Wasserman, L. "Estimating the Error Distribution of a Tap Sequence Without Ground Truth." *Proceedings of the 10th International Conference on Music Information Retrieval (ISMIR 2009),* (October 2009), pp. 297-302.

[3] Glus, B. "Further remarks on line segment curve-fitting using dynamic programming," *Communications of the ACM* 5:8, 1962.

[4] Honing, H. "When a Good Fit Is Not Good Enough: A Case Study on the *Final Ritard.*" *Proceedings of the 8th International Conference on Music Perception & Cognition,* 2004, pp. 510-513.

[5] Horner A. and Beauchamp J., "Piecewise Linear Approximation of Additive Synthesis Envelopes: A Comparison of Various Methods," *Computer Music Journal* 20:2, 1996.

[6] Kahn, A. *Kind of Blue: The Making of the Miles Davis Masterpiece.* Da Capo Press, 2001.

[7] Levene, H. "Robust tests for equality of variances." In Olkin, ed., *Contributions to Probability and Statistics*, I. Olkin, ed., Palo Alto: Stanford University Press, pp. 278-292.

[8] Wright, M. *Computer-Based Music Theory and Acoustics.* Ph.D. Thesis, Stanford University, CA, USA, March 2008.