(c) 2011 IEEE. Personal use of this material is permitted. Permission from IEEE must be obtained for all other users, including reprinting/ republishing this material for advertising or promotional purposes, creating new collective works for resale or redistribution to servers or lists, or reuse of any copyrighted components of this work in other works.

Published version: IEEE Signal Processing Letters, 18(3), 157-160, 2011 doi:10.1109/LSP.2010.2102018

Measuring the performance of beat tracking algorithms using a beat error histogram

Matthew E. P. Davies*, *Member, IEEE,* Norberto Degara and Mark D. Plumbley, *Member, IEEE*

***Abstract*—We present a new evaluation method for measuring the performance of musical audio beat tracking systems. Central to our method is a novel visualisation, the beat error histogram, which illustrates the metrical relationship between two qausi-periodic sequences of time instants: the output of beat tracking system and a set of ground truth annotations. To quantify beat tracking performance we derive an information theoretic statistic from the histogram. Results indicate that our method is able to measure performance with greater precision than existing evaluation methods and implicitly cater for metrical ambiguity in tapping sequences.**

I. Introduction

The research topic of audio beat tracking is well known within the music information retrieval community. Its aim is to recover a sequence of regular time instants from a musical input that are consistent with when a human listener might tap their foot [1]. While this problem has received much attention in terms of the development of beat tracking algorithms, e.g. [2], [3], [4] and comparative studies [5], far less effort has been placed on techniques used to measure performance. However evaluation is extremely important; without a meaningful measure of performance it is very difficult to assess the strengths and weaknesses of beat tracking algorithms and reliably compare them.

The basis of objective beat tracking evaluation is to compare two sequences of time instants: the output of a beat tracking algorithm and a sequence of ground truth annotations. The annotations are normally obtained by recording the tap times of a musical expert and then modifying them to correct any errors [6]. Given these two sequences, the role of the evaluation method is to provide a meaningful measurement of how well the beat locations "match" the annotations. The extent to which a match can be determined is based on two factors: temporal localisation and metrical level. For beats to be considered accurate they must be close in time to the annotations and tapped a tempo which is meaningful for the specific musical excerpt.

We do not expect the beat locations and annotations to coincide precisely at the same time instants. To account for this uncertainty most existing evaluation methods employ *tolerance windows*. These are placed around each ground truth annotation such that any beat falling within their range is considered accurate. While the size of the tolerance window can be calculated in absolute time (e.g. ±70 milliseconds [2]) or in proportion with the inter-annotation-interval, (e.g. ±20% [5]), the decision over their size is somewhat arbitrary. Using too narrow a window may fail to capture perceptually accurate forms of tapping, while too wide and performance may be overestimated

If we consider that for most pieces of music there isn't a single unambiguous tempo at which to tap the beat [5], the issue of metrical levels must be addressed by the evaluation method. If only a single ground truth sequence is provided without any information about which other metrical levels may be perceptually valid, then two options exist. The first is to consider only this interpretation to be valid and punish other interpretations even if they may be acceptable to a human listener. The second method is to adopt a heuristic approach where beats can be accurate if they are tapped at double or half the tempo of the annotations [3]. However merely allowing 2:1 and 1:2 ratios will only be appropriate for music with a 4/4 time-signature (i.e. four beats per bar); meaningful metrical levels for other time-signatures will also be punished.

Given the inherent limitations placed on beat tracking evaluation by using tolerance windows and pre-defined metrical relationships we propose a new approach able to contend with these issues. Our method is based on modelling the distribution of timing error between beats and annotations. We use this distribution, the *beat error histogram*, directly as an informative visualisation of beat tracking performance. From the histogram we show how different metrical interpretations can be observed. To provide a quantitative measure of beat tracking performance we propose an accuracy score which indicates the *information gain* a beat tracker provides over a uniformly distributed (i.e. completely unrelated) sequence of beat times compared to the annotations. In effect, we measure "how much better than random" the beats are.

Through simulations on an existing beat tracking database we demonstrate that our approach is able to capture cases of accurate tapping which are inaccurate using other methods. Furthermore, we can measure performance with greater precision than traditional tolerance window based methods, particularly in circumstances where tolerance window based approaches indicate 100% accuracy.

The remainder of this paper is structured as follows: in Section II we present the beat error histogram. This is followed by a description of the information gain measurement of performance in Section III. We present results in Section IV with conclusions in Section V.

Manuscript received___________

M.E.P. Davies and M.D. Plumbley are with Queen Mary University of London, School of Electronic Engineering and Computer Science, London E1 4NS, UK. (e-mail: matthew.davies@eecs.qmul.ac.uk, mark.plumbley@elec.qmul.ac.uk). N. Degara is with the Signal Theory and Communications Department, University of Vigo, Vigo, Spain. (email: ndegara@gts.tsc.uvigo.es).

II. Beat Error Histogram

A. Measuring Beat Error

To avoid the reliance on tolerance windows we formulate our evaluation method by measuring the timing error between beats and annotations. Assuming a sequence of B beats and J annotations, we notate the b^{th} beat, γ_b, and the j^{th} annotation, a_j. Comparing beat times to annotations we measure the timing error $\zeta_{\gamma|a}$ between each beat and the closest annotation,

$$\zeta_{\gamma|a}(b) = \begin{cases} \frac{\min_j(|\gamma_b - a|)}{\Delta_j} & \text{if} \quad j = J \quad \text{or} \quad \gamma_b \leq a_j \\ \frac{\min_j(|\gamma_b - a|)}{\Delta_{j+1}} & \text{if} \quad j = 1 \quad \text{or} \quad \gamma_b > a_j. \end{cases} \tag{1}$$

To contend with tempo changes, we normalise the timing error relative to the appropriate inter-annotation-interval (IAI), $\Delta_j = a_j - a_{j-1}$, depending on whether γ_b occurs before or after a_j. In this way, the timing error $\zeta_{\gamma|a}$ is bounded between -0.5 and 0.5 for all beats occurring within the range of the first to last annotations. If any beats occur more than half the IAI *before* the first or *after* the last annotation, these are mapped back into the range $[-0.5, 0.5]$ using modulo arithmetic.

If we consider an example where the beats are tapped at half the tempo of the annotations, then every other annotation will be close to a beat, however no timing error measurement will be made for the remaining annotations. Here, these “floating” annotations could occur at highly irregular locations and not affect the timing error. To contend with this situation, we follow the *two-way mismatch* procedure of Maher and Beauchamp [7] and form a second sequence of beat error, $\zeta_{a|\gamma}$, in which we measure the timing error between each annotation and the nearest beat,

$$\zeta_{a|\gamma}(j) = \begin{cases} \frac{\min_b(|a_j - \gamma|)}{\Delta_b} & \text{if} \quad b = B \quad \text{or} \quad a_j \leq \gamma_b \\ \frac{\min_b(|a_j - \gamma|)}{\Delta_{b+1}} & \text{if} \quad b = 1 \quad \text{or} \quad a_j > \gamma_b. \end{cases} \tag{2}$$

In this way, the *under-detection* of beats to annotations is transformed into the *over-detection* of annotations to beats. Henceforth we will refer to the timing error $\zeta_{\gamma|a}$ (beats compared to annotations) as the *forward beat error* and $\zeta_{a|\gamma}$ (annotations to beats) as the *backward beat error*.

B. Histogram

To visualise the behaviour of a beat tracking algorithm we determine the probability density function (pdf) for the forward and backward beat error sequences. Each pdf is estimated by calculating a K-bin histogram over the range of -0.5 to 0.5. Since K specifies how finely the beat error is quantised, it is important to select an appropriate number of bins. Too few (e.g. $K < 10$) and we may fail to adequately capture the shape of the distribution. Conversely having more bins than individual error measurements will mean some bins cannot be occupied and the resulting histogram will be too sparse. Through informal tests we found $K = 40$ to be sufficient to obtain a good estimate of the probability distribution of beat error for musical excerpts of at least 30 seconds. For the majority of existing test databases (e.g. [6], [3]) this is constraint is not problematic, however for very short sequences our method cannot currently be applied.

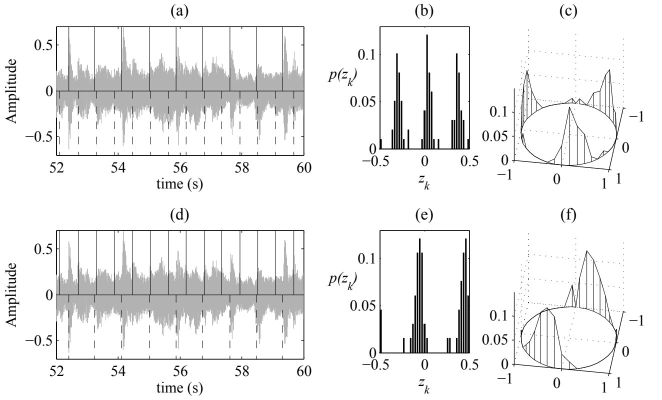

Fig. 1: Example forward and backward analysis. (a) annotations (solid), beats (dashed), (b) forward error histogram, (c) forward circular histogram. (d) annotations (dashed), beats (solid), (e) backward error histogram, (f) backward circular histogram.

Given K bins, $p_x(z_k)$ represents the estimated probability of bin k, such that the distribution of errors sum to unity, i.e. $\sum_{k=1}^{K} p_x(z_k) = 1$, where x refers to either the forward beat error, $\zeta_{a|\gamma}$, or the backward beat error, $\zeta_{\gamma|a}$. We calculate the bin centres, z_k, such that a beat error of zero will fall exactly in the middle of a histogram bin (not at the boundary between two histogram bins), with the same true of beat errors equivalent to -0.5 and 0.5. Plotted on a linear scale from -0.5 to 0.5, this means that the first and last bins are half the width of the others. In subsequent calculations the contents of these two bins are summed together and treated as a single bin. Organising the histogram bin centres in this way enables a simple mapping onto the unit circle, with circular bin centres, $c_k = (2\pi k/K) - \pi$. Example forward and backward beat error histograms are shown in Fig. 1.

Visual inspection of the histograms highlights the two main properties when comparing beat sequences: metrical relationship and temporal localisation. In Fig. 1(a) there are three beats for every two annotations and hence three main peaks in the forward beat error histogram. Similarly in the backward beat error histogram (Fig. 1(d)) we find two main peaks, consistent with the two annotations occurring for every three beats (for sound examples see [8]). In general, if a regular metrical relationship exists between the two sequences it can be observed as the ratio of the number of modes in the forward error histogram to the number in the backward error histogram.

In terms of localisation of beats and annotations, we can see that the peaks in the histograms are not centred on a beat error of zero. Inspection of Fig. 1(a) shows that the estimated beats are consistently “late” compared to the annotations. Given the histogram visualisation, any systematic offsets between the beats and annotations can be identified and hence corrected.

III. Information Gain

While the beat error histogram is an informative visualisation we also wish to extract a numerical measurement of beat tracking accuracy. Towards this aim we consider two extremes of beat tracking performance. First, where the beat locations are identical to the annotations, we would obtain a delta function in both the forward and backward beat error histograms. Considering the worst case of beat tracking, where the beats and annotations are entirely unrelated, we should expect near uniform distributions of beat error. This can arise in one of two ways, either if the beats are sampled from a uniform distribution, or if they are regular but tapped at a non-meaningful tempo (e.g. 109 bpm compared to 100 bpm). This leads to *tempo drift* where occasional beats are close annotations and considered accurate, but no relationship exists.

Our aim is for the numerical accuracy to meaningfully reflect these two extremes of beat tracking while accounting for tempo drift. To this end we could measure the variance of the beat error histogram. However, if we re-examine the examples in Figs. 1 (b) and (d), which are both multi-modal distributions, the resulting high variance would not reflect the perceptual accuracy of the beats.

An alternative is to look for a description of the *peakiness* of the pdf of beat error, which can be determined by measuring the entropy of the histogram. However, instead of using the entropy directly, we use a related quantity, the *information gain*. We calculate the divergence between the empirical beat error pdf of a given beat tracking algorithm and a uniform pdf indicative of the theoretically worst beat tracker. In effect we are measuring the dependence between the two sequences. If they are unrelated we will have low information gain; with high information gain if a relationship exists.

We find the information gain, I_x in terms of the Kulback-Leibler divergence between each beat error distribution with estimated mass probability $p_x(z_k)$ and the uniform histogram with K bins of height $1/K$ as,

$$\begin{aligned} I_x &= \sum_{k=1}^{K} p_x(z_k)\log_2(\frac{p_x(z_k)}{\frac{1}{K}}) && (3)\\ &= \sum_{k=1}^{K} p_x(z_k)\log_2(p_x(z_k)) + \log_2(K) && (4)\\ &= \log_2(K) - H(p_x(z_k)) && (5)\end{aligned}$$

where $H(p_x(z_k))$ is the entropy of the estimated beat error distribution of the beat tracking algorithm under evaluation,

$$H(p_x(z_k)) = -\sum_{k=1}^{K} p_x(z_k)\log_2(p_x(z_k)). \qquad (6)$$

Given that we have two beat error histograms to analyse, derived from $\zeta_{\gamma|a}$ and $\zeta_{a|\gamma}$, we extract the both the forward and backward information gains, $I_{\gamma|a}$ and $I_{a|\gamma}$ respectively. To prevent overestimating the information gain given by the beat tracker, which could arise if very few beats were compared to many annotations, we keep the lower information gain, such that $I = \min(I_{\gamma|a}, I_{a|\gamma})$. The information gain is measured in *bits* and is lower and upper bounded by $0 \leq I \leq \log_2(K)$.

Because the entropy calculation in (6) is invariant to the ordering of the bins, any beat-relative shift of a histogram will have the same information gain. Therefore tapping the "off-beat" in reference to the annotations will have the same information gain as beats which are "in-phase".

IV. Results

To illustrate the properties of the information gain evaluation method we compare it to the performance scores given by four existing evaluation methods.

PScore: beat accuracy is measured by finding the sum of a time-limited cross-correlation between impulse trains representing the beats and the annotations. A tolerance window of ±20% of the median IAI specifies the region around each annotation for which beats can be accurate [5].

Cemgil: beat accuracy is calculated by measuring the timing error between each annotation and the temporally closest beat; the timing error is evaluated on a Gaussian error function which assigns low scores for beats which are poorly localised to annotations [9].

CMLc: beat accuracy is found as the ratio of the longest continuously correct segment to the length of the excerpt for beats tapped at the correct metrical level; each beat must fall within a ±17.5% tolerance window around the annotation and the previous beat be within the previous tolerance window [4].

AMLt: as CMLc, however the continuity requirement is relaxed and beats may be tapped in anti-phase (the "off-beat"), at twice or half the metrical level of the annotations [4].

On an annotated beat tracking database containing 222 excerpts [6] we measure the performance of a beat tracking algorithm (available as a plugin for Sonic Visualiser[1]) using each of the four evaluation methods described above and compare these to the information gain. To visualise the differences in performance, scatter plots are shown in Fig. 2. Following the reproducible research model [10] we make available code to regenerate the figures in this paper [8].

In each of the scatter plots for the fixed tolerance window methods, see Fig. 2(a), (b) and (d), we can observe clusters of points which score near to 100% under each evaluation method. However, within these clusters there are a range of information gain values. This highlights a limitation of using a fixed tolerance window. For many excerpts the beats are sufficiently accurate to fall within the range of the tolerance windows, but no further distinction can be made between them as the *limit* of accuracy has been reached. Without repeated recalculation of performance over a range of sizes of tolerance window [4], any comparison between beat tracking algorithms (and discrimination between 100% accurate systems) is constrained by the choice of tolerance window. However if the output of one beat tracking algorithm is better localised to the annotations than another this will appear in the beat error histograms and result in a higher information gain. Used in this way, information gain can reveal accuracy beyond the resolution to the tolerance windows, and provide additional discrimination between beat tracking algorithms.

The **Cemgil** score, in Fig. 2(c), which does not use a fixed tolerance window, appears to be strongly correlated with information gain. Note, for both methods it is very unlikely to obtain "perfect" performance. For the information gain this would require all beat error measurements to fall within a single bin of the histogram and for the **Cemgil** score the beat times and annotations would have to be identical. Although the

[1] http://isophonics.net/QMVampPlugins

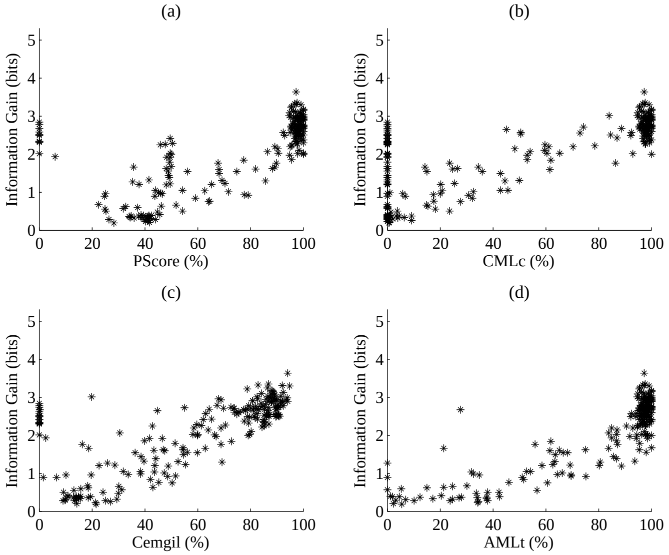

Fig. 2: Scatter plots showing information gain scores for the beat tracker output against the following evaluation methods: (a) PScore; (b) CMLc; (c) Cemgil; and (d) AMLt.

two scores are related, the **Cemgil** score is only calculated for a single metrical level. Therefore any beat sequences tapped at other tempi will be punished in proportion with the number of beats and annotations which not well-localised, and any beat sequences tapped on the off-beat will score zero. We can observe this behaviour in Fig. 2(c) where several excerpts score a high information gain but are among the least accurate for the **Cemgil** score. A similar pattern can be observed for the tolerance window approaches, including the **AMLt** method, which allows tapping at double or half the annotated metrical level and the off-beat. Here other meaningful metrical relationships exist beyond the scope of the allowed metrical levels, e.g. the “two-against-three” case in Fig. 1.

When evaluating beat tracking systems using existing methods we should pay particular attention to beat tracking accuracy scores of either 100% or 0%. In the former case, information gain can be used to discriminate between the systems, allowing us to find the beat sequence which is best localised to the annotations. In the latter case, a high information gain can indicate if a relationship exists between the beats and annotations beyond those permitted by predefined rules; alternatively a low information gain can confirm that the sequences are indeed unrelated.

V. Conclusions

We have presented a new evaluation method for measuring the performance of beat tracking algorithms based on the generation of two beat error histograms, one representing the comparison of beat times to annotations and the other comparing annotations to beats. From these histograms we calculate an information theoretic measure of performance based on the peakiness of the histograms to indicate the level of dependence between the two sequences.

While we have demonstrated our approach can contend with the limitations of existing tolerance window based methods, it has certain surprising properties which arise from estimating performance from a single ground truth annotation sequence. By explicitly choosing *not* to make any assumptions about other likely metrical levels it is possible (although somewhat unlikely in practice) to achieve a high information gain from unusual relationships (e.g. 5 beats for every 3 annotations). However this would be observable by inspection of the histograms. Also, by treating beat-relative shifts as equivalent, beats which are consistently early or late can also appear accurate. Although this may be deemed problematic, there is evidence to suggest human tappers behave in this way while still perceiving the beat [11]; and again such behaviour could be identified in the beat error histograms.

In future work we will investigate how to extend our model to exploit multiple annotation sequences, e.g. by weighting the contribution of regions of the beat error histogram which correspond to acceptable metrical relationships and offsets. In addition we plan to explore the dependence between beat tracking algorithms by comparing their output without ground truth, e.g. as in [12].

Beyond its use an evaluation method we hope that our visualisation can be used as a diagnostic tool for investigating the qualitative behaviour of beat tracking systems towards enhancing performance of future beat tracking systems.

VI. Acknowledgements

This work was partially supported by EPSRC Grants EP/G007144/1 and EP/E045235/1 and Spanish MCYT TEC2009-14414-C03-03. We would like to thank the anonymous reviewers for their comments and recommendations.

References

[1] D. P. W. Ellis, “Beat tracking by dynamic programming,” *Journal of New Music Research*, vol. 36, no. 1, pp. 51–60, 2007.

[2] S. Dixon, “Automatic extraction of tempo and beat from expressive performances,” *Journal of New Music Research*, vol. 30, pp. 39–58, 2001.

[3] A. P. Klapuri, A. Eronen, and J. Astola, “Analysis of the meter of acoustic musical signals,” *IEEE Transactions on Audio, Speech and Language Processing*, vol. 14, no. 1, pp. 342–355, 2006.

[4] M. E. P. Davies and M. D. Plumbley, “Context-dependent beat tracking of musical audio,” *IEEE Transactions on Audio, Speech and Language Processing*, vol. 15, no. 3, pp. 1009–1020, 2007.

[5] 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*, vol. 36, no. 1, pp. 1–16, 2007.

[6] S. Hainsworth, “Techniques for the automated analysis of musical audio,” Ph.D. dissertation, Dept. of Engineering, Cambridge University, 2004.

[7] R. C. Maher and J. W. Beauchamp, “Fundamental frequency estimation of musical signals using a two-way mismatch procedure,” *Journal of the Acoustical Society of America*, vol. 95, no. 4, pp. 2254–2263, 1994.

[8] M. E. P. Davies and N. Degara. (2010) Supplimentary material. [Online]. Available: http://www.gts.tsc.uvigo.es/∼ndegara/beat_evaluation.zip

[9] A. T. Cemgil, B. Kappen, P. Desain, and H. Honing, “On tempo tracking: Tempogram representation and Kalman filtering,” *Journal Of New Music Research*, vol. 28, no. 4, pp. 259–273, 2001.

[10] P. Vandewalle, J. Kovacevic, and M. Vetterli, “Reproducible research in signal processing - what, why and how,” *IEEE Signal Processing Magazine*, vol. 26, pp. 37–47, 2009.

[11] S. Dixon, W. Goebl, and E. Cambouropoulos, “Perceptual smoothness of tempo in expressively performed music,” *Music Perception*, vol. 23, no. 3, pp. 195–214, 2006.

[12] R. B. Dannenberg and L. Wasserman, “Estimating the error distribution of a single tap sequence without ground truth,” in *Proceedings of 10th International Conference on Music Information Retrieval*, 2009, pp. 297–302.

Following the reproducible research model [Vandewalle, et al., *IEEE Signal. Process. Mag.*, vol. 26, pp. 37–47, 2009], audio examples and source code to regenerate the figures in this paper are available at the Sound Software Code Repository http://code.soundsoftware.ac.uk/projects/beh