TOWARDS AUTOMATICALLY CORRECTING TAPPED BEAT ANNOTATIONS FOR MUSIC RECORDINGS

Jonathan Driedger[1], **Hendrik Schreiber**[2], **W. Bas de Haas**[1], **and Meinard Müller**[2]
[1] Chordify, [2] International Audio Laboratories Erlangen
jonathan@chordify.net

ABSTRACT

A common method to create beat annotations for music recordings is to let a human annotator tap along with them. However, this method is problematic due to the limited human ability to temporally align taps with audio cues for beats accurately. In order to create accurate beat annotations, it is therefore typically necessary to manually correct the recorded taps in a subsequent step, which is a cumbersome task. In this work we aim to automate this correction step by "snapping" the taps to close-by audio cues—a strategy that is often used by beat tracking algorithms to refine their beat estimates. The main contributions of this paper can be summarized as follows. First, we formalize the automated correction procedure mathematically. Second, we introduce a novel visualization method that serves as a tool to analyze the results of the correction procedure for potential errors. Third, we present a new dataset consisting of beat annotations for 101 music recordings. Fourth, we use this dataset to perform a listening experiment as well as a quantitative study to show the effectiveness of our snapping procedure.

1. INTRODUCTION

Identifying the time positions of beats in music recordings has been a core task in the Music Information Retrieval (MIR) community for a long time. Irrespectively of whether the goal is to evaluate beat tracking algorithms or to train new data-driven models for beat detection, it is necessary to have accurate annotations that describe the temporal locations of beats in music recordings. The beat positions of a music recording are often loosely defined as the time instances where a human would tap along when listening to it [6]. A straightforward approach to create beat annotations is therefore to record these taps—for example by using a specialized audio player software like *Sonic Visualizer* [5], which allows annotators to tap on a key of the keyboard. This method was used, for example, in [20, 21, 25, 26]. However, this procedure is problematic due to the limited ability of humans to accurately align their taps with acoustic cues that are associated with beats such as instrument onsets, percussive sound events, or chord changes [29, 30]. Perception literature indicates that, depending on the complexity of a recording, the onset times of two tones must differ by less than 40 milliseconds such that they may be perceived as being temporally aligned [19]. This means that when sonifying human-made taps with a click track, a click and an audio cue for a beat have to both fall into an interval of at most 40 milliseconds such that the click could be perceived as accurately representing the beat position. [1] This is often not the case as is illustrated in Figure 1. In Figure 1a, we see a short spectrogram excerpt of the song "T'envoler" by Paul Daraiche (item 006 in our dataset, see Table 1 for the YouTube link). One can observe the vertical spectral structures originating from the piano onsets in the song's intro. The human-made taps are visualized in Figure 1b. They roughly coincide with the audio cues, but precede them by about 80 milliseconds most of the time. To obtain more accurate beat annotations, several approaches have been used in the past. One way is to manually correct the taps of a human annotator in a subsequent step [20,21]. However, manual corrections are cumbersome to perform since often every single tap has to be corrected individually—usually in a drag&drop fashion using tools like Sonic Visualizer. Another approach, which has been used in [15, 24], is to compute an initial

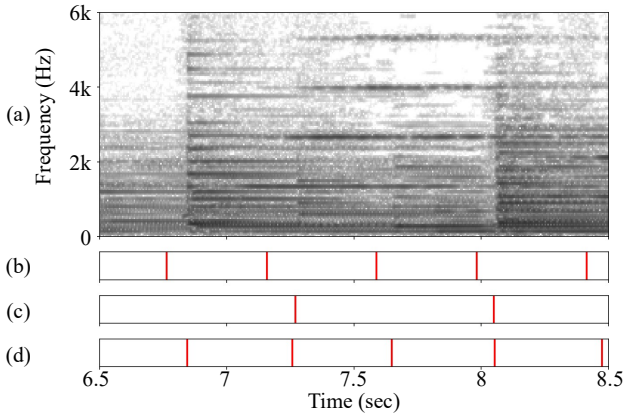

Figure 1. **(a)** Excerpt of the spectrogram for item 006 from our dataset, **(b)** taps by the annotator, **(c)** beat positions as estimated by [4], **(d)** automatically corrected taps.

[1] Note that the 40 milliseconds constitute an upper bound. Depending on the quality of the audio cue this threshold may be significantly lower.

estimate of the beat positions using a beat tracking algorithm. Beat trackers such as [3, 14] actively aim to "snap" potential beat candidates to close-by audio cues in order to create accurate results. In Figure 1c, we see the beat positions as estimated by *madmom*, a Python library featuring a state-of-the-art beat tracker [3, 4]. The estimated beats are well aligned with the audio cues visible in the spectrogram. However, we also see that only every other beat has been captured. Correcting these kinds of errors made by beat tracking algorithms can be just as cumbersome as manually correcting the taps of a human annotator.

In this paper, we propose a new way of creating beat annotations. Our idea is to mostly automate the manual correction of human-made taps by using the concept of snapping beat candidates to audio cues. The intuition is that the taps made by an annotator constitute good beat candidates that are located in close proximity to the actual beat positions. Therefore, snapping them to nearby audio cues should accurately correct the vast majority of taps. This is visualized in Figure 1d, where the automatically corrected taps are aligned with the audio cues in the spectrogram. In this paper, we explore this simple idea in a systematic fashion. First, in Section 2, we model the automatic correction procedure mathematically. Then, in Section 3, we propose a novel visualization that can serve as a tool to reveal rhythmically challenging sections in music recordings as well as potential errors made by the snapping procedure. Section 4 is dedicated to experiments. In Section 4.1, we apply our proposed annotation strategy to create a dataset of beat annotations for 101 music recordings from YouTube. In Section 4.2, we then discuss the results of a listening experiment that shows that human listeners perceive the corrected taps as more accurate than the original taps. In Sections 4.3, we finally conduct a small study that investigates the effect of using either the original or the corrected taps as ground truth for the quantitative evaluation of different beat tracking algorithms. For the purpose of reproducibility, we made our Python implementations (snapping procedure and visualizations) as well as the dataset of beat annotations along with YouTube identifiers publicly available at [13].

2. PROPOSED PROCEDURE

In this section, we formalize our procedure for the automatic correction of human-made taps. We start by explaining the core ideas in Section 2.1 and discuss choices of specific processing steps in Section 2.2.

2.1 Basic Principle

The goal of our proposed procedure is to correct the human-made taps by snapping them to nearby audio cues in the music recording. Our fundamental assumption is that each of the taps indicates the rough position of a beat such that the "real" beat position can be found in close temporal proximity. Given the music recording $x : \mathbb{Z} \to \mathbb{R}$ (Figure 2a), we first derive an *activation curve* $a : \mathbb{Z} \to \mathbb{R}$ (Figure 2b) that is sampled at a sampling rate of $f_s \in \mathbb{R}^+$

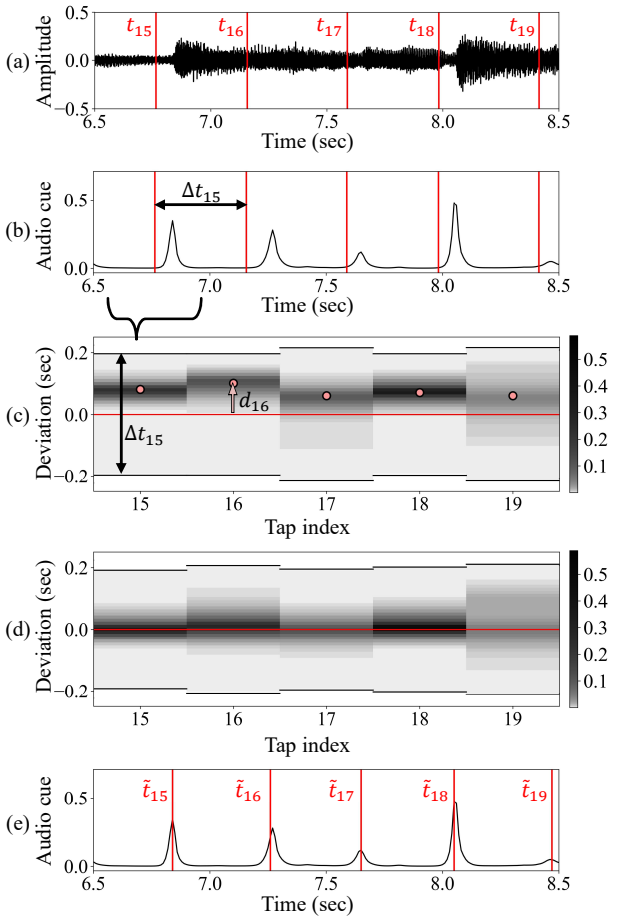

Figure 2. Proposed tap correction procedure. **(a)** Music recording x with taps t. **(b)** Activation curve a. **(c)** Deviation function $\mathcal{D}_t$ with envelope reflecting the inter-tap-intervals, tap positions indicated by red line, and deviation sequence d indicated by light-red dots. **(d)** Deviation function $\mathcal{D}_{\tilde{t}}$ with corrected taps $\tilde{t}$ indicated by red line. **(e)** Activation curve a with corrected taps $\tilde{t}$ indicated in red.

(we use f_s=100 Hertz as suggested in [4, 12, 17]). An activation curve a can be seen as a function whose values $a(n)$ reflect how likely it is that there is a beat present in the music recording x at time n/f_s. We discuss different choices for activation curves in Section 2.2. Along with x and a, we are also given the *sequence of taps* $t = [t_0, \ldots, t_{M-1}]$ with $t_m \in \mathbb{Z}$. Each tap t_m indicates that the human annotator tapped at time t_m/f_s. In our example in Figure 2b one can see that the taps are not well aligned with the peaks in the activation curve a.

With the activation curve and the taps, we now compute the *deviation function* $\mathcal{D}_t : \mathbb{Z} \times [0 : M-1] \to \mathbb{R}$ by

$$\mathcal{D}_t(n, m) := w_m(n)\, a(t_m + n)$$

for $n \in \mathbb{Z}, m \in [0 : M-1]$. Here, $w_m : \mathbb{Z} \to \mathbb{R}$ is a Hann window centered around zero, whose length is defined by the *inter-tap-interval*

$$\Delta t_m := t_{m+1} - t_m$$

for $m \in [0 : M-2]$ and we define $\Delta t_{M-1} := \Delta t_{M-2}$ (such that Δt_m is defined for all taps). The reason for using a

window function is to effectively implement our assumption that the taps are located in close temporal proximity to the actual beat positions. The more temporal distance between a tap and a high activation value, the less likely it is that the activation value reflects the actual beat the tap was meant to represent. In the visualization of $\mathcal{D}_t$ seen in Figure 2c, the length of w_m is indicated with two additional black lines in each column of $\mathcal{D}_t$. This "envelope" of $\mathcal{D}_t$ serves as a visual representation of the individual inter-tap-intervals. In our example, the envelope does not exhibit any significant variation across the shown taps which indicates that the annotator tapped with an almost constant tempo. However, when looking at the individual activation maxima in each column of $\mathcal{D}_t$—which can be found at deviations between 70 and 110 milliseconds—it becomes obvious that the inaccuracy of the human-made taps is not just a constant offset but varies over time.

In the next step we compute the *deviation sequence* $d = [d_0, \ldots, d_{M-1}]$, $d_m \in \mathbb{Z}$, that indicates the individual corrections we will apply to each tap t_m (Figure 2c). There are several options of how to derive d from $\mathcal{D}_t$ which we discuss in Section 2.2. From t and d we finally compute the *automatically corrected taps* $\tilde{t} = [\tilde{t}_0, \ldots, \tilde{t}_{M-1}]$ by

$$\tilde{t}_m := t_m + d_m \ .$$

Based on $\tilde{t}$ we now can also compute the deviation function $\mathcal{D}_{\tilde{t}}$ (Figure 2d). Note that in this visualization, all high activation values are now centered around a deviation of zero. Furthermore, the envelope of $\mathcal{D}_{\tilde{t}}$ does not differ significantly from the envelope of $\mathcal{D}_t$ which indicates that the inter-tap-intervals in $\tilde{t}$ are similar to those in the original tap sequence t. This can be verified when plotting $\tilde{t}$ on top of the activation curve a where the taps now accurately align with the peaks (Figure 2e).

2.2 Technical Realization

In traditional music signal processing, a common choice for activation curves are *novelty curves*, see for example [2, 7, 27]. These functions are designed to reflect sudden temporal changes in a music recording's spectrogram, which are typically caused by percussive sound events such as instrument onsets. As beats often go along with these kinds of sound events it is reasonable to use a novelty curve as activation curve in our tap correction procedure. We denote this activation curve by a^{nov} and the resulting deviation function for taps t by $\mathcal{D}_t^{\text{nov}}$.

Another option is using activation curves based on data-driven models. Recently, models that were trained to transform spectrogram representations of music recordings into activation curves have significantly improved the quality of state-of-the-art beat tracking algorithms [16]. For example in [4], a *deep neural network* (DNN) using a bidirectional long-short-term memory architecture is trained on a large collection of beat-annotated music recordings for that task. It was shown in [4] that the peaks in activation curves derived using this model align well with beats in the underlying music recordings. Using it in our procedure is therefore

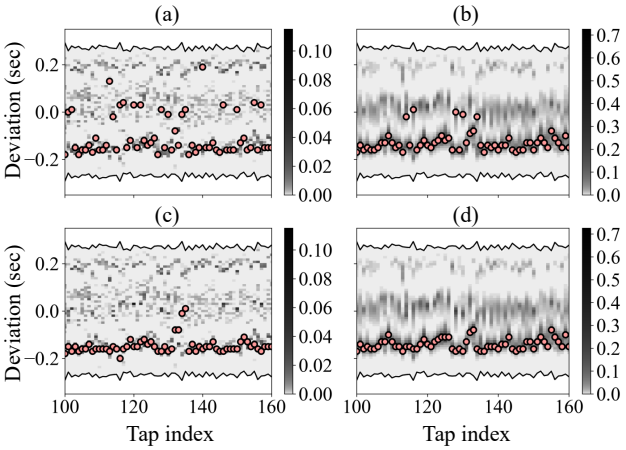

Figure 3. Excerpts of different deviation functions and deviation sequences for item `011`. **(a)** $\mathcal{D}_t^{\text{nov}}$ with $d^{\max}$ derived from it. **(b)** $\mathcal{D}_t^{\text{DNN}}$ with $d^{\max}$ derived from it. **(c)** $\mathcal{D}_t^{\text{nov}}$ with d^{con} derived from it. **(d)** $\mathcal{D}_t^{\text{DNN}}$ with d^{con} derived from it.

reasonable as well (we use the implementation freely available in [3]). We denote the resulting activation curves by a^{DNN} and the deviation function by $\mathcal{D}_t^{\text{DNN}}$, respectively.

We also have several choices concerning the derivation of the deviation sequence d from $\mathcal{D}_t$. A straight-forward way is to simply pick the deviation that yields the highest activation value in each column of $\mathcal{D}_t$ as

$$d_m^{\max} := \underset{n}{\operatorname{argmax}}\, \mathcal{D}_t(n, m) \ .$$

While this method considers every tap individually, it is also possible to incorporate some contextual information into the derivation by defining

$$d^{\text{con}} := \underset{[d_0,\ldots,d_{M-1}]}{\operatorname{argmax}}\, \mathcal{D}_t(d_0, 0) \prod_{m=1}^{M-1} \mathcal{D}_t(d_m, m)\, \mathcal{T}(d_{m-1}, d_m),$$

with $\mathcal{T} : \mathbb{Z} \times \mathbb{Z}$ being a transition function defined by

$$\mathcal{T}(i, j) := e^{-\lambda|i-j|}$$

with $i, j \in \mathbb{Z}$. The sequence d^{con} can be derived using dynamic programming. The idea, inspired by [14, 22], is that subsequent human taps are unlikely to drastically vary in their deviation from the actual beat. To reflect this, the use of the transition function $\mathcal{T}$ makes it unlikely to have large deviation jumps from one tap to the next (we use λ=0.1 in our experiments). Furthermore, this method also allows us to correct *non-event beats* [18], meaning taps for which no cue in the music recording exists. In this case, the activation curve shows no salient values around the tap and $\mathcal{T}$ favors a constant deviation until there are clear cues in the activation curve again.

Figure 3 shows the two types of deviation functions $\mathcal{D}_t^{\text{nov}}$ and $\mathcal{D}_t^{\text{DNN}}$ in combination with the two methods for deriving the deviation sequences $d^{\max}$ and d^{con}. We use item `011` in our dataset as an example (see Table 1). It is a rather old recording featuring singing voice, acoustic guitar, mouth-organ, and piano. Comparing $\mathcal{D}_t^{\text{nov}}$ in Figure 3a/c to $\mathcal{D}_t^{\text{DNN}}$ as seen in Figure 3b/d, we can observe

item	YouTubeID	artist	song title
006	I3gHugP6bPE	Paul Daraiche	T'envoler
009	hRFLv29K__o	La Sonora Dinamita	Escandalo
011	M7u5SdjDSQQ	The Lovin' Spoonful	Daydream
025	8jsFGdeWNPo	Nicky Jam	Juegos Prohibidos
040	1zrxnqejwCk	Los Tigres Del Norte	Mañanitas Tapatias
046	ebZZpVFUQDY	Green Valley	Relaja
048	B_e7QbWc5mI	Orthodox Celts	Rocky Road To Dublin

Table 1. List of dataset items used throughout this paper. The YouTube videos can be found by using the URL `www.youtube.com/watch?v=[YouTubeID]`.

that $\mathcal{D}_t^{\text{nov}}$ is noisier while the structures seen in $\mathcal{D}_t^{\text{DNN}}$ are smoother and more salient. This becomes obvious when comparing the two $d^{\max}$ in Figure 3a and b. While the $d^{\max}$ based on $\mathcal{D}_t^{\text{nov}}$ jumps back and forth between deviations of about -0.2 and $+0.2$ seconds, $d^{\max}$ for $\mathcal{D}_t^{\text{DNN}}$ is more stable, showing only a few jumps between tap indices 115 and 135. The strength of using contextual information in the computation of the deviation sequence is visible in Figure 3c, where we see d^{con} based on $\mathcal{D}_t^{\text{nov}}$. Here, similar as in Figure 3b, most deviation values cluster around -0.2 seconds, except for a short passage of deviation 0 around tap index 135. This is caused by a single very strong activation value which does not coincide with a beat position in the recording. In the deviation function $\mathcal{D}_t^{\text{DNN}}$ this spurious activation is not present and the deviation sequence d^{con} in Figure 3d does not show prominent jumps. When listening to the sonified automatically corrected taps based on this deviation sequence, one can hear that they are in fact very accurate. Overall, we made similar observations for the vast majority of songs in our dataset. For this reason, we chose the combination of $\mathcal{D}_t^{\text{DNN}}$ with d^{con} in our subsequent experiments and also refer to them by just $\mathcal{D}_t$ and d for the sake of simplicity.

3. ANALYSIS OF AUTOMATED CORRECTIONS

Although the method introduced in the previous section is capable of automatically correcting the vast majority of taps, there is still potential for error. To find these errors efficiently, visualizing the deviation functions $\mathcal{D}_t$ and $\mathcal{D}_{\tilde{t}}$ can give helpful insights into the automatic correction process, point to problematic sections in music recordings, and reveal anomalies in the human-made taps. In Figure 4, we show several examples.

Figure 4a depicts the deviation functions of the original and automatically corrected taps for item `009`. This latin american song features strong and steady rhythmic pulses, which is reflected in the activation values visible in $\mathcal{D}_t$. In each column of $\mathcal{D}_t$, there is basically only one high activation value. The deviation sequence d nicely captures this train of high activations, which leads to a very clean deviation function $\mathcal{D}_{\tilde{t}}$. Note that the inter-tap-intervals in $\mathcal{D}_{\tilde{t}}$ are more regular than in $\mathcal{D}_t$, which can be seen by the envelope of $\mathcal{D}_{\tilde{t}}$ being less noisy than the one of $\mathcal{D}_t$. Based on this visualization, it is rather safe to assume that the corrected taps are accurate with virtually no errors. One thing noteworthy about this example is that d has a fairly constant offset of about -90 milliseconds, meaning that almost all original taps were about 90 milliseconds behind the beat. This was caused by technical problems in the process of recording the taps of the human annotator, which caused delays between the physical taps and the registered times.

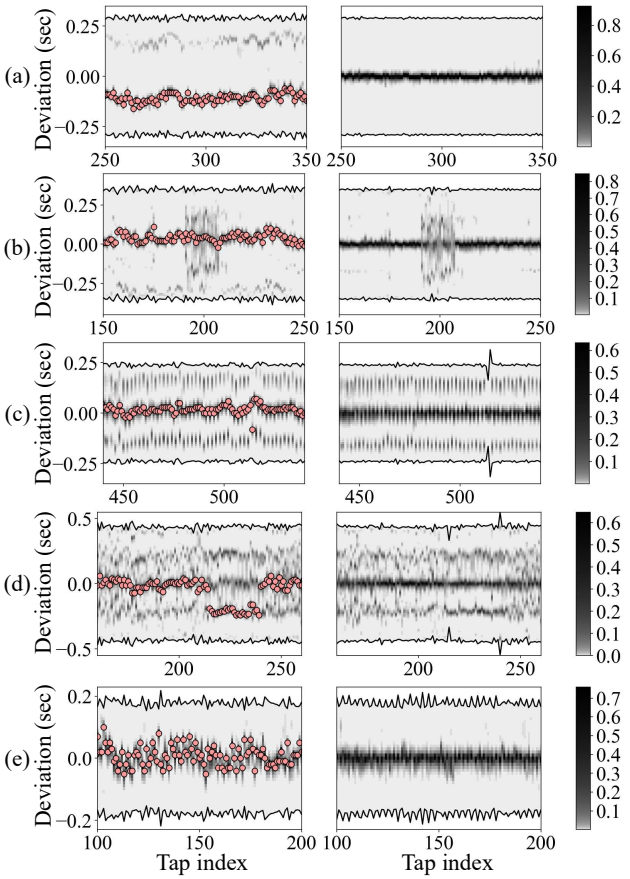

Figure 4. Deviation functions $\mathcal{D}_t$ (left) and $\mathcal{D}_{\tilde{t}}$ (right) for excerpts of different items from our proposed dataset. **(a)** Item `009` (2:25 to 3:23), **(b)** item `025` (2:12 to 3:22), **(c)** item `048` (3:30 to 4:18), **(d)** item `046` (2:21 to 3:49), **(d)** item `040` (0:37 to 1:13).

In Figure 4b, we see the deviation functions for item `025`. This hip hop song again has a very clear and prominent beat, which is reflected in the activations in $\mathcal{D}_t$. However, around the 200^{th} beat, the song has a short part without any percussions. This manifests in $\mathcal{D}_t$ as a blurry section, which is caused by low and diffuse activation values. Since this indicates that there are fewer audio cues that the procedure can utilize to correct the original taps, such sections should be manually inspected after the automatic correction step. In this particular example the inspection showed that no manual corrections were necessary.

As a third example, we see the deviation functions for item `048` in Figure 4c. In the last part of this Irish folk song the bass drum plays a swing-like rhythm. This pattern causes the activation structures as seen in $\mathcal{D}_t$ with strong activations around deviation zero and weaker ones at about ± 200 milliseconds for every other tap. In the computation of the deviation sequence d, this lead to an error at tap in-

dex 510, where the tap was incorrectly snapped to one of the activations caused by the rhythmic ornaments. This single tap, misplaced by the correction procedure, can be easily detected in our visualization, since it caused a distinct structure in the envelope of $\mathcal{D}_{\tilde{t}}$, where it manifested in a "lightning" pattern.

A similar error can be seen in Figure 4d, which shows the deviation functions of an excerpt from item `046`. This acoustic song featuring vocals, guitar, and keyboard is pretty challenging for beat tracking due to the lack of strong audio cues for beats, as can be seen by the rather noisy activations in $\mathcal{D}_t$. As in the previous example, the used rhythmic pattern, this time played by the guitar, causes strong activations at non-beat positions. In the computation of the deviation sequence d, these lead to a section of about 30 taps that were incorrectly snapped to these offbeat guitar accents rather than the actual beats. This can be seen by d being shifted to a deviation of about -0.2 seconds from tap index 210 to 240. The visualization of d allowed us to easily locate and understand this problematic passage, which could then be corrected manually in a post-processing step.

As a final example, Figure 4e shows the deviation functions of an excerpt from item `040`. This Polka-like song has a $3/4$ time signature, but the third beat in each bar is consistently played late. [2] The human annotator tapped through the song in a rather straight fashion, not explicitly reflecting this rhythmic pattern. The automatic correction procedure then aligned each tap with the closest instrument onset, resulting in unevenly spaced corrected taps. This is visible as regular spike pattern in the envelope of $\mathcal{D}_{\tilde{t}}$. Whether the corrected taps reflect the "true" beat positions is dependent on whether one sees the delayed note onsets as part of the rhythm or mere ornamentation. Either way, this interesting example was easily revealed by our visualization of the deviation functions.

4. EXPERIMENTS

In this section, we evaluate our proposed procedure. We first introduce in Section 4.1 a dataset of beat annotations which we created using our correction procedure. Then, in Section 4.2, we discuss the results of a listening experiment to show that the corrected taps are actually perceived as being more accurate than the original taps. Finally, in Section 4.3, we show how the choice of annotation used as ground truth influences the evaluation of beat tracking algorithms.

4.1 The Dataset

In order to show the usefulness of our proposed tap correction procedure in a real-world annotation scenario, we applied it to create a new dataset of beat annotations. To this end, we selected 101 different music recordings available on YouTube. [3] This collection, which consists of about 7.25 hours of music in total, comprises a variety of different genres, recording conditions and instrumentations. Similar to [1, 23], we decided to use music recordings from YouTube to ensure reproducibility of our results. For each of these recordings, a musically experienced annotator tapped along to create the tap sequences t using Sonic Visualizer, adding up to 41.011 individual taps. The sequences of automatically corrected taps $\tilde{t}$ were then computed for each recording using the method described in Section 2. Each sequence $\tilde{t}$ was then manually inspected by the first author using Sonic Visualizer. In this step, a total of 331 incorrect taps were identified across 54 of the 101 sequences and corrected manually. We denote the resulting *fully corrected tap sequences* by $\tilde{t}'$.

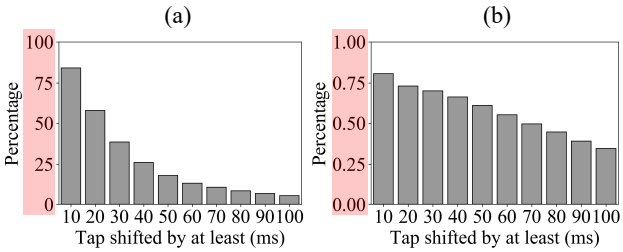

Figure 5. Percentage of the 41.011 original taps that were shifted in the two consecutive automatic and manual correction steps. Note the different scales of the vertical axes. **(a)** Creating $\tilde{t}$ from t, **(b)** creating $\tilde{t}'$ from $\tilde{t}$.

Figure 5 summarizes the distribution of shifts applied to the individual taps in the two consecutive correction steps. In Figure 5a we can see that about 25% of the original taps were shifted by 40 milliseconds or more by the automatic correction procedure. This means that in case these corrections would have been done manually, about one in four taps, would have been shifted—even when assuming that smaller inaccuracies where taps were misaligned with audio cues by less than 40 milliseconds would not have been considered. To create the fully corrected taps on the other hand, less than 1% of the 41.011 original taps were shifted at all, see Figure 5b. The shifts applied in the manual correction step were equally distributed between very small and larger shifts which can be seen by the rather constant slope of the graph.

The dataset containing all three annotations t, $\tilde{t}$, and $\tilde{t}'$ for each recording as well as metadata and the respective YouTube links is available at [13].

4.2 Listening Experiment

With the creation of the fully corrected taps $\tilde{t}'$, the main question is whether the applied corrections actually make the beat annotations perceptually more accurate—and therefore better. To answer this, we conducted a listening experiment. For each of the 101 recordings in our dataset, we created two new recordings by sonifying the taps t as well as the fully corrected taps $\tilde{t}'$ as a click track and superimposed each of them on the original mu-

[2] This observed rhythmic pattern is rather unusual for the style of the song but commonly found in Viennese Waltz.

[3] The dataset is part of a Chordify project that assesses the quality of automatically generated annotations in a large scale industry setting. The selected recordings reflect a random sample of songs used on Chordify, weighted by their popularity on the service.

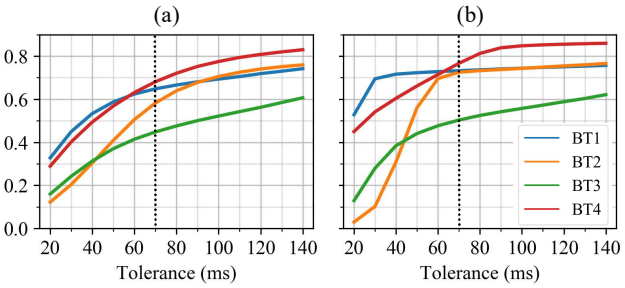

Figure 6. Average beat F-measures on our dataset for four different beat tracking algorithms when using different ground truth annotations. **(a)** Original taps t, **(b)** fully corrected taps $\tilde{t}'$.

sic recording. Note that we chose to use the fully corrected taps $\tilde{t}'$ rather than the automatically corrected taps $\tilde{t}$ because the difference between $\tilde{t}$ and $\tilde{t}'$ is very small and we did not want the participants to focus on the few noticeable mistakes made by the automatic correction procedure. Three musically trained people took part in the experiment, none of them being one of the authors. Each participant was presented with the 101 pairs of recordings and asked to pick the recording with the more accurate click track from each pair. The order in which the two recordings of a pair were presented was random and the participants did not know whether the clicks they heard were the original taps t or the fully corrected taps $\tilde{t}'$. They were able to listen to each recording for as long and as often as they liked before making their decision. Additionally, they had the opportunity to give comments. The complete set of answers and comments can be found at [13]. Looking at all 303 given answers individually (three participants times 101 pairs), 89% of the time the participants found the fully corrected taps $\tilde{t}'$ to be more accurate than the original taps t. For 72% of the 101 pairs, all three participants even consistently perceived $\tilde{t}'$ to be more accurate than t. Having a closer look at the participants' answers and comments, it turned out that in many instances where a participant chose the original taps to be more accurate than the fully corrected taps, the two click tracks were perceived as being very similar (*"Both are about the same quality, IMO."*). Furthermore, some of the comments also indicate that sometimes there was also a degree of personal preference involved in the decision (*"...[the click track] lags like a proper gospel drummer."*). Overall, the results show that the fully corrected taps $\tilde{t}'$ are commonly perceived as more accurate than the original taps t.

4.3 Quantitative Study

As a final experiment, we were interested in the effects of using either the original taps t or the fully corrected taps $\tilde{t}'$ as ground truth for a quantitative evaluation of beat tracking algorithms. We investigated four different algorithms: The *Queen Mary beat tracker* (BT1) [10], the *librosa beat tracker* (BT2) [14], the *Aubio beat tracker* (BT3) [8, 9], and our own implementation of [4] (BT4). Note that the madmom beat tracker [3,4], which would have an intrinsic advantage since our tap correction procedure is built upon the same activation curve, is not among them. For each of the four beat tracking algorithms, we computed the *beat F-measure* [11], a popular beat tracking evaluation measure, for all recordings in our dataset. We did this two times, once taking the original taps t as ground truth and once the fully corrected taps $\tilde{t}'$, see Figure 6a and 6b, respectively. An important parameter in the computation of the beat F-measure is the tolerance, which determines the maximal temporal distance between an estimated beat and a ground truth beat such that the estimate can be considered correct. In Figure 6, we see that for a very large tolerance (140 milliseconds and above) it basically makes no difference whether we use t or $\tilde{t}'$ as ground truth. This makes sense when recalling that most of the corrections applied were smaller than 100 milliseconds (see Section 4.1). However, this changes when considering a smaller, and hence more realistic tolerance. The default tolerance as implemented in *mir_eval* [28], the Python library we used for this evaluation, is 70 milliseconds, indicated by dotted black lines in Figure 6. At this tolerance, the computed beat F-measures differ noticeably depending on the ground truth. For example, BT4 achieves an average beat F-measure of 0.68 when the original taps t are used as ground truth, but 0.77 when the fully corrected taps $\tilde{t}'$ are used. The difference is even more prominent when comparing BT1 and BT2. Here, BT1 scores much better (0.65) than BT2 (0.58) when comparing the two algorithms based on the original taps. However, when using the fully corrected taps as ground truth, the algorithms perform nearly identically (both 0.73). For smaller tolerances, the differences between the individual algorithms become even more salient. For example, at a tolerance of 30 milliseconds and using the original taps as ground truth, BT1 and BT2 differ in beat F-measure by 0.25 (scoring 0.45 and 0.20, respectively), while differing by 0.59 (scoring 0.69 and 0.10, respectively) on the fully corrected taps. This shows that it can make a substantial difference for quantitative evaluations of beat tracking algorithms whether the underlying ground truth annotations are "only" human-made taps or corrected ones.

5. CONCLUSIONS

In this paper we proposed and formalized a simple procedure for correcting tapped beat annotations by automatically "snapping" the tap positions to cues in the underlying music recording. Furthermore, we proposed a visualization that can help identifying errors made by the correction procedure as well as rhythmically interesting passages in music recordings. Finally, we created a new dataset for which we showed that beat annotations corrected with our procedure are perceived as being more accurate and that using them for the quantitative evaluation of beat tracking algorithms may significantly impact the evaluation results. The last observation motivates us to apply our proposed correction procedure to beat annotated datasets commonly used in the MIR community. We believe that our proposed visualization could help in identifying incorrectly annotated recordings and therefore getting a more realistic view on the performance of beat tracking algorithms.

6. ACKNOWLEDGMENTS

We would like to thank Leigh Smith and Sebastian Böck for our discussions at ISMIR 2018 that sparked the ideas for this contribution. Furthermore, we thank Jeroen Bransen for the discussions about the algorithmic details of the procedure, Jeffrey van Rossum for creating the original taps for the recordings in the dataset, as well our listening experiment participants Tijmen Ruizendaal, Matúš Tejiščák, and Bart Laan. Hendrik Schreiber and Meinard Müller are supported by the German Research Foundation (DFG MU 2686/10-1). The International Audio Laboratories Erlangen are a joint institution of the Friedrich-Alexander-Universität Erlangen-Nürnberg (FAU) and Fraunhofer Institut für Integrierte Schaltungen IIS.

7. REFERENCES

[1] S. Balke, C. Dittmar, J. Abeßer, K. Frieler, M. Pfleiderer, and M. Müller. Bridging the gap: Enriching YouTube videos with jazz music annotations. *Frontiers in Digital Humanities*, 2018.

[2] J. P. Bello, L. Daudet, S. Abdallah, C. Duxbury, M. Davies, and M. Sandler. A tutorial on onset detection in music signals. *IEEE Transactions on Speech and Audio Processing*, 13(5):1035–1047, 2005.

[3] S. Böck, F. Korzeniowski, J. Schlüter, F. Krebs, and G. Widmer. madmom: A new python audio and music signal processing library. In *Proceedings of the ACM Conference on Multimedia Conference*, pages 1174–1178, Amsterdam, The Netherlands, 2016.

[4] S. Böck, F. Krebs, and G. Widmer. Joint beat and downbeat tracking with recurrent neural networks. In *Proceedings of the 17th International Conference on Music Information Retrieval (ISMIR)*, pages 255–261, New York City, United States, 2016.

[5] 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*, pages 1467–1468, Firenze, Italy, October 2010.

[6] E. Clarke. Rhythm and timing in music. *The Psychology of Music*, 2:473–530, 12 1999.

[7] N. Collins. A comparison of sound onset detection algorithms with emphasis on psychoacoustically motivated detection functions. In *AES Convention 118*, Barcelona, Spain, 2005.

[8] M. E. P. Davies, P. Brossier, and M. D. Plumbley. Beat tracking towards automatic musical accompaniment. In *Proceedings of the Audio Engineering Society 118th Convention*, Barcelona, Spain, May 2005.

[9] M. E. P. Davies and M. D. Plumbley. Causal tempo tracking of audio. In *Proceedings of the 5th International Conference on Music Information Retrieval (ISMIR)*, Barcelona, Spain, October 2004.

[10] 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.

[11] S. Dixon. Automatic extraction of tempo and beat from expressive performances. *Journal of New Music Research*, 30:39–58, 2001.

[12] S. Dixon. Evaluation of the audio beat tracking system BeatRoot. *Journal of New Music Research*, 36:39–50, 2007.

[13] J. Driedger, H. Schreiber, W. B. de Haas, and M. Müller. Accompanying material: Towards automatically correcting tapped beat annotations for music recordings. `https://github.com/chordify/tapcorrect`.

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

[15] V. Eremenko, E. Demirel, B. Bozkurt, and X. Serra. Audio-aligned jazz harmony dataset for automatic chord transcription and corpus-based research. In *Proceedings of the 19th International Society for Music Information Retrieval Conference (ISMIR)*, pages 483–490, Paris, France, September 2018.

[16] M. Fuentes, B. McFee, H. C. Crayencour, S. Essid, and J. P. Bello. Analysis of common design choices in deep learning systems for downbeat tracking. In *Proceedings of the 19th International Society for Music Information Retrieval Conference (ISMIR)*, pages 106–112, Paris, France, September 2018.

[17] A. Gkiokas and V. Katsouros. Convolutional neural networks for real-time beat tracking: A dancing robot application. In *Proceedings of the 18th International Society for Music Information Retrieval Conference (ISMIR)*, pages 286–293, Suzhou, China, October 2017.

[18] P. Grosche, M. Müller, and C. S. Sapp. What makes beat tracking difficult? A case study on Chopin Mazurkas. In *Proceedings of the 11th International Conference on Music Information Retrieval (ISMIR)*, pages 649–654, Utrecht, The Netherlands, 2010.

[19] S. Handel. *Listening: An Introduction to the Perception of Auditory Events*. A Bradford book. MIT Press, 1993.

[20] A. Holzapfel, M. E. P. Davies, J. R. Zapata, J. Oliveira, and F. Gouyon. Selective sampling for beat tracking evaluation. *IEEE Transactions on Audio, Speech and Language Processing (TASLP)*, 20(9), 2012.

[21] A. Holzapfel, F. Krebs, and A. Srinivasamurthy. Tracking the "odd": meter inference in a culturally diverse music corpus. In *Proceedings of the 15th International Society for Music Information Retrieval Conference (ISMIR)*, pages 425–430, Taipei, Taiwan, October 2014.

[22] F. Krebs, S. Böck, and G. Widmer. An efficient state-space model for joint tempo and meter tracking. In *Proceedings of the 16th International Society for Music Information Retrieval Conference (ISMIR)*, pages 72–78, Málaga, Spain, 2015.

[23] P. López-Serrano, C. Dittmar, and M. Müller. Finding drum breaks in digital music recordings. In *Proceedings of the International Symposium on Computer Music Modeling and Retrieval (CMMR)*, Porto, Portugal, 2017.

[24] U. Marchand and G. Peeters. Swing ratio estimation. In *Proceedings of the International Conference on Digital Audio Effects (DAFx)*, Trondheim, Norway, 2015.

[25] M. Mauch, C. Cannam, M. E. P. Davies, S. Dixon, C. Harte, S. Kolozali, D. Tidhar, and M. Sandler. OMRAS2 metadata project 2009. In *Late-breaking session at the 10th International Society for Music Information Retrieval Conference (ISMIR)*, Kobe, Japan, 2009.

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

[27] M. Müller. *Fundamentals of Music Processing*. Springer Verlag, 2015.

[28] C. Raffel, B. McFee, E. J. Humphrey, J. Salamon, O. Nieto, D. Liang, and D. P. W. Ellis. MIR_EVAL: A transparent implementation of common MIR metrics. In *Proceedings of the 15th International Society for Music Information Retrieval Conference*, pages 367–372, Taipei, Taiwan, October 2014.

[29] B. H. Repp. Sensorimotor synchronization: A review of the tapping literature. *Psychonomic Bulletin & Review*, 12(6):969–992, 2005.

[30] C. Weiß, V. Arifi-Müller, T. Prätzlich, R. Kleinertz, and M. Müller. Analyzing measure annotations for western classical music recordings. In *Proceedings of the 17th International Conference on Music Information Retrieval (ISMIR)*, pages 517–523, New York, USA, 2016.