GENERALIZED METRICS FOR SINGLE-F0 ESTIMATION EVALUATION

Rachel Bittner*, Juan J. Bosch*
Spotify, USA
*Equal contribution

ABSTRACT

Single-f_0 estimation methods, including pitch trackers and melody estimators, have historically been evaluated using a set of common metrics which score estimates frame-wise in terms of pitch and *voicing* accuracy. "Voicing" refers to whether or not a pitch is active, and has historically been regarded as a *binary* value. However, this has limitations because it is often ambiguous whether a pitch is present or absent, making a binary choice difficult for humans and algorithms alike. For example, when a source fades out or reverberates, the exact point where the pitch is no longer present is unclear. Many single-f_0 estimation algorithms select a threshold for when a pitch is active or not, and different choices of threshold drastically affect the results of standard metrics. In this paper, we present a refinement on the existing single-f_0 metrics, by allowing the estimated voicing to be represented as a continuous likelihood, and introducing a weighting on frame level pitch accuracy, which considers the energy of the source producing the f_0 relative to the energy of the rest of the signal. We compare these metrics experimentally with the previous metrics using a number of algorithms and datasets and discuss the fundamental differences. We show that, compared to the previous metrics, our proposed metrics allow threshold-independent algorithm comparisons.

1. INTRODUCTION

Single-f_0 estimation algorithms, including pitch trackers and melody or bass extraction algorithms, predict fundamental frequency (f_0) over time for an audio file. However, there can be time intervals where there is no (target) f_0 value present, for example during silent regions. To account for this, single-f_0 estimation methods additionally estimate the *voicing* over time - i.e. when a given frame contains an active pitch or not. Choosing when the estimated voicing should be active/voiced (1) or inactive/unvoiced (0) often involves choosing a threshold on a confidence value. Single-f_0 estimation algorithms are evaluated by comparing the accuracy of the estimated f_0 and voicing sequence against a reference f_0 and voicing sequence. The choice of threshold to estimate voicing has a critical effect on the resulting metrics; the threshold is often treated as a hyperparameter and is chosen on a validation set. Any confidence information used to determine voicing is discarded and not considered in the evaluation metrics.

The perceptual salience of a pitch is affected by a number of factors, including the volume, the frequency content, the duration and the presence of interference from other sources [12, 18, 21, 27]. In some cases, the brain can perceive a pitch even when the f_0 is not physically present, for example when one short time segment in the middle of a longer pitch sequence is set to be silent [9]. The effect of these factors can be different for each listener, making the task of "objectively" determining if a pitch is present or not a difficult one. Additionally, in polyphonic mixtures, a pitch can be masked by other sources. In the current metrics, algorithms are equally penalized for mistakes on salient and non-salient f_0 values.

We propose a generalization of the existing metrics which (1) allows an algorithm to report voicing as a continuous value (between 0 and 1), and (2) allows frames to be weighted by a reward, which more heavily penalizes mistakes in frames where the energy of the source producing the f_0 is high compared to the rest. These changes remove the need for making a strict decision on whether or not a pitch is present, allowing threshold-independent algorithm comparisons, and allow an optional weighting to be added to reflect frame importance. We also show that when the provided voicing is binary, the proposed metrics are equivalent to the existing metrics. The proposed metrics are to be seen as complementary to the classic ones, which remain useful for measuring performance for applications where a binary threshold is needed in practice. However, binary thresholds are not needed for a number of applications, including pitch informed source separation or melodic similarity.

Additionally, generalized metrics would also help mitigate non-uniformity in decisions made when annotating datasets (e.g. inclusion of delays and reverb as part of the annotation or not), by rewarding correct pitch estimations proportionally to the energy of the pitched signal to be detected. Furthermore, they provide information about the confidence of the estimators, which is useful for many applications.

For reproducibility, the code used for this in this paper is available online [1] .

[1] `github.com/juanjobosch/continuousf0eval`

2. VOICING DETERMINATION

Historically, single-f_0 estimation methods need to determine whether a given frame contains a pitch or not. To perform this binary decision, algorithms have commonly used a (static or dynamic) threshold on e.g. energy, salience or pitch likelihood [1, 3, 13–15, 23]. For instance, melody extraction algorithms may exploit pitch contour salience distributions and use heuristics [24], or a threshold on melody contour probabilities produced by a discriminative model [4, 7]. Durrieu et al. [14] first perform source separation on the melodic source, and subsequently estimate the energy of the separated signal frame by frame; frames with energy above a threshold are determined to be voiced. The threshold is empirically selected such that 99.95% of the leading instrument energy is contained in voiced frames. Fuentes et al. [15] also use an energy threshold (of -12dB) on a low-pass filtered separated melody signal. The ideal threshold typically depends on the difference in intensity between melody and accompaniment.

Some methods bypass the use of an explicit threshold and deal with voicing estimation using a classifier, for instance by adding an "unvoiced" class to the set of possible pitch outputs [2]. Other approaches model singing voice detection separately from pitch estimation, and even try to exploit information from neighboring frames (e.g. with LSTMs) for making a decision on the presence of melody on a given frame [17, 23]. Finally, some of the state of the art algorithms provide a measure of confidence on their estimations. However, traditional evaluation metrics do not consider this information.

3. CLASSIC EVALUATION METRICS

Pitch estimation methods have commonly been evaluated using metrics derived from information retrieval, commonly focused on pitch-related accuracy and seldom consider voicing [11, 16]. Melody extraction algorithms are evaluated using similar metrics to pitch estimation, but voicing also takes an important role.

Symbol	Description
n	sample index $\in \{0, \dots, N-1\}$
f_n	reference frequency (Hz) at sample n
v_n	reference voicing $\in \{0, 1\}$ at sample n
r_n	pitch estimation reward $\in [0, 1]$ at sample n
$\hat{f}_n$	estimate frequency (Hz) at sample n
$\hat{v}_n$	estimate voicing $\in [0, 1]$ at sample n

Table 1. Definition of symbols.

For melody estimation in particular, several metrics are commonly used in the literature [22, 26]. Raw Pitch Accuracy (RPA) and Raw Chroma Accuracy (RCA) measure pitch-related estimation quality. Let the reference and estimate f_0 and voicing sequences be defined as in Table 1. RPA measures the percentage of melody frames in the reference for which the estimated pitch is considered correct (usually within half a semitone of the reference). RCA also measures pitch accuracy, but both estimated and reference pitches are mapped into one octave, forgiving octave mistakes.

$$\texttt{RPA} = \frac{\sum_{n=0}^{N-1} v_n \mathcal{T}_{\hat{f}_n, f_n}}{\sum_{n=0}^{N-1} v_n}, \ \texttt{RCA} = \frac{\sum_{n=0}^{N-1} v_n \mathcal{O}_{\hat{f}_n, f_n}}{\sum_{n=0}^{N-1} v_n} \quad (1)$$

where the "correct pitch" indicator function is defined as:

$$\mathcal{T}_{\hat{f}_n, f_n} = \begin{cases} 1 & |d_s(\hat{f}_n, f_n)| \leq 0.5 \\ 0 & |d_s(\hat{f}_n, f_n)| > 0.5 \end{cases} \quad (2)$$

and the difference d_s between two frequency values in semitones is defined as:

$$d_s(\hat{f}_n, f_n) = 12 \log_2 \left(\frac{\hat{f}_n}{f_n} \right) \quad (3)$$

Similarly, the "correct chroma" indicator function is defined as:

$$\mathcal{O}_{\hat{f}_n, f_n} = \begin{cases} 1 & |d_o(\hat{f}_n, f_n)| \leq 0.5 \\ 0 & |d_o(\hat{f}_n, f_n)| > 0.5 \end{cases} \quad (4)$$

and the single-octave pitch difference d_o is defined as:

$$d_o(\hat{f}_n, f_n) = d_s(\hat{f}_n, f_n) - 12 \left\lfloor \frac{d_s(\hat{f}_n, f_n)}{12} + 0.5 \right\rfloor \quad (5)$$

Voicing estimation is evaluated with Voicing Recall rate (VR) and Voicing False Alarm rate (VFA). VR measures the percentage of frames labeled as voiced in the reference which are also estimated as voiced by the algorithm. On the other hand, VFA measures the percentage of frames labeled as un-pitched in the reference that are mistakenly estimated as melody frames by the algorithm.

$$\texttt{VR} = \frac{\sum_{n=0}^{N-1} \hat{v}_n v_n}{\sum_{n=0}^{N-1} v_n}, \ \texttt{VFA} = \frac{\sum_{n=0}^{N-1} \hat{v}_n (1 - v_n)}{\sum_{n=0}^{N-1} (1 - v_n)} \quad (6)$$

Finally, Overall Accuracy (OA) is used as a single aggregate measure to evaluate algorithms, as it accounts for both pitch and voicing estimation accuracy. In particular, OA measures the percentage of frames that were correctly labeled in terms of both pitch and voicing.

$$\texttt{OA} = \frac{1}{N} \sum_{n=0}^{N-1} v_n \hat{v}_n \mathcal{T}_{\hat{f}_n, f_n} + (1 - v_n)(1 - \hat{v}_n) \quad (7)$$

In order to allow each of the metrics to give insights about different aspects of methods, the traditional evaluation methodology allows algorithms to report "negative" pitch values, which are only considered for the computation of RPA and RCA. This was an attempt to evaluate voicing estimation performance separately from pitch estimation performance, and therefore make the result of RPA

Metric	**Classic** ($r_n = v_n$, binary $\hat{v}_n$)	**Generalized** (continuous r_n and $\hat{v}_n$)
RPA	average pitch accuracy in voiced frames	weighted average pitch accuracy in voiced frames
RCA	average chroma accuracy in voiced frames	weighted average chroma accuracy in voiced frames
VR	fraction of voiced frames estimated as voiced	average voicing likelihood in voiced frames
VFA	fraction of unvoiced frames estimated as voiced	average voicing likelihood in unvoiced frames
OA	fraction of frames with correct voicing and pitch	weighted average correctness of each frame

Table 2. Description of the meaning of the metrics in the classic and generalized cases. Note that two possible cases are not described: binary $\hat{v}_n$ with a continuous reward r_n, and continuous $\hat{v}_n$ with $r_n = v_n$.

and RCA independent of the voicing estimation. However, many algorithms do not actually report negative pitches, and furthermore due to the inner functioning of some methods (e.g. Melodia [24]), increasing the number of reported pitches (either positive or negative) not only has an effect on voicing estimation accuracy but also on pitch accuracy.

Other metrics have been proposed to give further insights, such as the continuity of the correctly estimated pitches (either in pitch or chroma), which is relevant for tasks such as automatic transcription, source separation or visualization [8]. Metrics related to user satisfaction have also been studied in the context of melody extraction: different kind of errors have a different impact in the quality perceived when users listen to synthesized melodies that have been extracted automatically [20]. However, most single-f_0 estimation literature does not consider the influence of the energy of the signal under study (or its relation to the accompaniment) in the evaluation. Some exceptions [6, 25] present an evaluation of pitch salience functions, which are commonly correlated to the energy of the signals. Bosch et al. [8] also study the influence of the predominance of the melody over the accompaniment for different algorithms in the context of symphonic music, and monophonic pitch estimators have been evaluated in the presence of different noise levels [16, 28]. However, in the classic single-f_0 estimation metrics, all frames contribute equally to the results, even though in many cases the presence or absence of a (melody) pitch may be unclear for both humans and algorithms.

4. GENERALIZED METRICS

This section presents a generalization of the traditional metrics, in order to deal with the previously introduced limitations: voicing estimates must be binary, and all frames receive equal importance. The proposed metrics (1) allow algorithms to report voicing $\hat{v}_n$ as a *continuous* rather than a binary quantity, representing the likelihood that the frame is voiced, and (2) optionally weight the pitch accuracy in voiced frames using a reward $r_n \in [0, 1]$, allowing mistakes in less important frames to count less than mistakes in important frames.

In the following metrics, we require that (1) $r_n = 0$ if and only if $v_n = 0$, (2) $v_n = 0$ if $f_n = 0$ and (3) $\hat{v}_n = 0$ if $\hat{f}_n = 0$. Note that we may have $\hat{f}_n \neq 0$ and $\hat{v}_n = 0$, allowing the metrics to score pitch accuracy when voicing mistakes are made.

Equation 8 presents the proposed generalization of RPA and RCA, which aggregate the pitch/chroma accuracy proportional to the reward r_n. This makes the generalized metrics more forgiving on unimportant frames, and more demanding on important frames in comparison to previous metrics.

$$\mathtt{RPA} = \frac{\sum\limits_{n=0}^{N-1} r_n \mathcal{T}_{\hat{f}_n, f_n}}{\sum\limits_{n=0}^{N-1} r_n}, \; \mathtt{RCA} = \frac{\sum\limits_{n=0}^{N-1} r_n \mathcal{O}_{\hat{f}_n, f_n}}{\sum\limits_{n=0}^{N-1} r_n} \qquad (8)$$

Generalized versions of VR and VFA remain the same as in Equation 6, however $\hat{v}_n$ need not be binary. In both cases, VR is simply the average of $\hat{v}_n$ in voiced frames ($v_n = 1$), and similarly VFA is the average of $\hat{v}_n$ in unvoiced frames ($v_n = 0$).

Finally, we propose a generalized version of OA in Equation 9, which scores voiced frames proportionally to $\hat{v}_n$, weighted by r_n, and scores unvoiced frames proportionally to $1 - \hat{v}_n$. Let $V = \sum\limits_{n=0}^{N-1} v_n$, the number of voiced frames in the reference annotations. The generalized OA becomes:

$$\mathtt{OA} = \frac{V \frac{\sum_{n=0}^{N-1} r_n \hat{v}_n T_{\hat{f}_n, f_n}}{\sum_{n=0}^{N-1} r_n} + (N - V) \frac{\sum_{n=0}^{N-1} (1 - v_n)(1 - \hat{v}_n)}{\sum_{n=0}^{N-1} (1 - v_n)}}{N} \qquad (9)$$

In this generalized OA, voicing "mistakes" are penalized according to the confidence of the estimator, which is softer than in the binary case where mistakes are "all or nothing". When $r_n = v_n$ (equal reward in all voiced frames) and $\hat{v}_n$ is binary, each of the generalized metrics is equivalent to the metrics defined in the binary case. For RPA, RCA, VR and VFA the equivalence is straightforward. For OA, substituting r_n by v_n, plugging in the given equation for V, and simplifying the resulting quantity shows equivalence. Table 2 gives summaries of the metrics in the classic and generalized cases.

4.1 Behavior on Artificial Examples

Figure 1 shows the behavior of the proposed metrics for a few simple examples. For instance, the different results obtained in plots (a), (b), (e) and (f) show that if the pitches are correct, VR and OA get the best results with highly confident estimations. Plot (a) also shows that errors in the

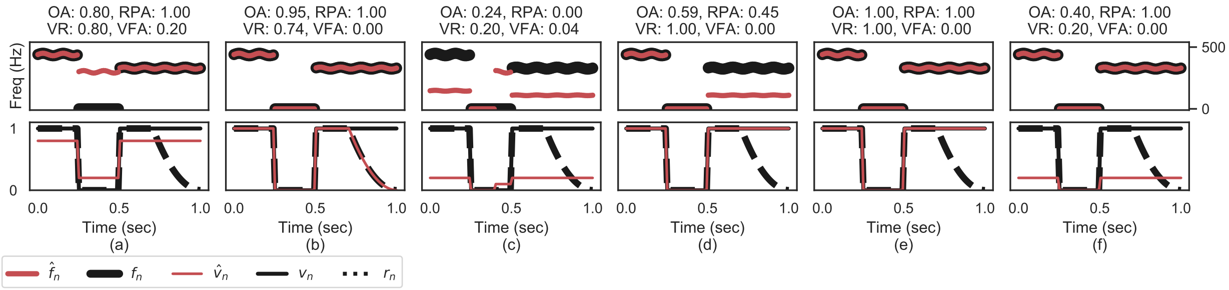

Figure 1. Artificial examples of different combinations of f_n, r_n, $\hat{f}_n$, and $\hat{v}_n$, and the behavior of the generalized metrics.

voicing estimation are less penalized if the estimate confidence is low. In this example, VFA is relatively low (0.2), while if the algorithm had used a very low threshold to determine a binary voicing value $\hat{v}_n$, VFA would be equal to 1 (the worst possible score), since all unvoiced frames would have been estimated as voiced. Plot (d) shows the effect of having perfect voicing estimation but incorrect pitch estimation – the estimator is penalized in OA and RPA. Plot (c) shows a completely wrong pitch estimation, which gets a small score for voicing recall, due to the low estimated confidence. In the same plot, VFA is very low due to the correctly identified unvoiced frames, and the errors between 0.4 and 0.5 s are not heavily penalized due to the low reported confidence.

5. COMPUTING PITCH ESTIMATION REWARDS

In order to create the pitch estimation reward (r_n) for single f_0 datasets, we propose the computation of Root-Mean-Square (RMS) energy in frames over time. The first step is to compute the RMS of the source producing the f_0 (RMS$_{f_0}$) and the RMS of the mixture (RMS$_m$) in frames. The second step deals with a frame-wise normalization, in order to obtain a reward signal: r_n = (RMS$_{f_{0n}}$)/(max(RMS$_{f_0}$)+RMS$_{m_n}$), where n corresponds to the index of the frame. In frames where there is no pitch annotation ($f_n = 0$), we set $r_n = 0$, as illustrated in Figure 2.

A mismatch between the energy of the melodic source and the voicing derived from pitch annotations (if the f_0 is non zero) could happen due to several factors. One of them is the fact that there may be energy due to a melodic instrument but actually no pitch, for instance in transient percussive sounds, or with unpitched vocal sounds (e.g., many of the consonants). Another possible factor is that the procedure followed during the annotation did not consider echos or reverberation, while they might be clearly present in the signal.

5.1 Isolated Sources

For datasets where isolated sources are available, we can simply compute the frame-wise RMS of the signals over time. For instance, in a melody extraction dataset, we would use the RMS of the source playing the melody in each frame to derive RMS$_{f_0}$. Note that this is compatible with multiple melody definitions, even allowing different instruments to play the melody sequentially in a given music excerpt [5,8]. RMS$_m$ is computed from the instruments which are not playing the melody pitch in each frame, and the reward computed following the methodology from section 5. A simpler case corresponds to monophonic pitch estimation datasets, where RMS$_m$ is zero, so the raw reward in each frame is equal to the RMS$_{f_0}$, normalized by its maximum RMS value in the example.

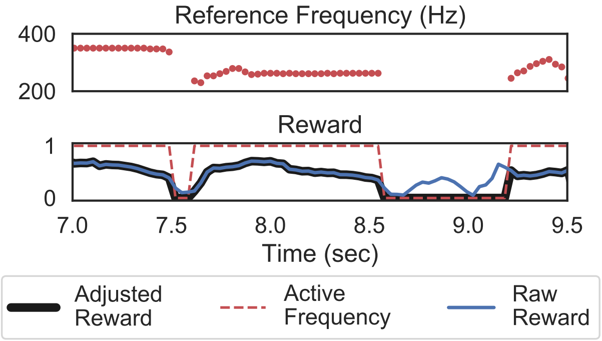

Figure 2. The reference frequency and the estimated reward values. Frames where no reference frequency is provided may have non-zero reward estimates (in blue) - in these cases the reward is set to 0 (in black).

5.2 Sources in Polyphonic Mixtures

When isolated sources are not available, it is more difficult to compute RMS$_{f_0}$ and RMS$_m$. For these cases, we propose the use of pitch-informed source separation [8, 14] in order to obtain an estimate of the energy of the source and accompaniment. We test the effectiveness of this approach using the iKala dataset by comparing the difference in the reference reward when computing it using isolated vocals, and using the results of pitch informed source separation on the mixture signal.

Figure 3 shows the results of "Melody-A" (see Section 6, for all metrics, with the confidence computed both using source separation and in the ideal case (having access to the isolated sources). As we can observe, VR and VFA have the same values, and OA, RPA, and RCA present some small differences but which are statistically significant, according to a paired t-test ($\alpha = 0.05$). However, in Figure 3 (right) we see that for each metric, the distribution of differences in comparison to "Melody-B" is very

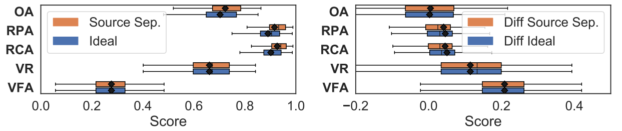

Figure 3. (Left) Metrics for "Melody-A" on iKala for confidence computed using both source separation and in the ideal case. (Right) The difference in score between "Melody-A" and "Melody-B" per track for both confidence measures on iKala.

similar when using confidence computed on both the ideal case and with source separation. This suggests that using source separation as a proxy to get the confidence measure would not have an impact on the ranking of algorithms, and therefore the methodology proposed would be useful to compare different algorithms. We leave the improvement of pitch-informed source separation for obtaining a better reward, i.e. more similar to the values obtained if the isolated sources were available, as future work.

6. PROPOSED METRICS ON REAL DATA

In order to show the behavior of the metrics with real data and algorithms, we create variants of four established algorithms: two monophonic pitch estimators "Pitch-A" (based on CREPE [16]) and "Pitch-B" (based on pYIN [19]) and two melody extraction algorithms "Melody-A" (based on Deep Salience [3]) and "Melody-B" (based on Melodia [24]). Note that the main objective is not to actually evaluate/compare these algorithms, but show the behaviour and give further insights about the proposed metrics. Therefore the arbitrarily taken decisions about the estimators such as the normalization, or using default parameters, should not be regarded as important.

In order to test our metrics, we need each algorithm to produce a continuous voicing estimate $\hat{v}_n$. "Pitch-A" and "Melody-A" predict confidence values as part of the algorithm, which we use directly as $\hat{v}_n$. "Pitch-B" and "Melody-B" do not directly predict confidence values, but determine which frames are voiced and unvoiced using thresholds on signals computed internally. We derive a value of $\hat{v}_n$ for these algorithms using normalized versions of these signals (the maximum probability of the pitch candidates for "Pitch-B", and the contour confidence measure for "Melody-B"). Note that not all algorithms currently provide a confidence value as an output, but all of them determine voicing at some level, and the steps used to make this decision can typically be used to create a measure of voicing confidence.

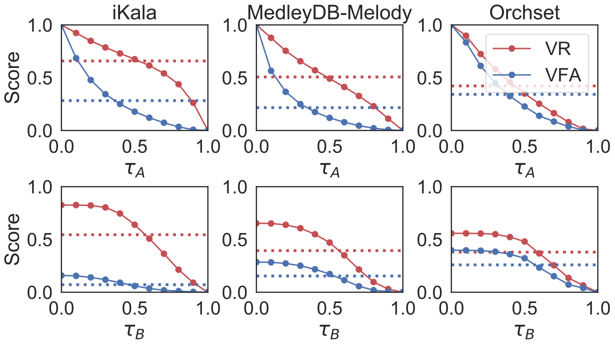

Figure 4. Classic voicing metrics (VR - red, VFA - blue) as a function of threshold. Dashed horizontal lines show the value of the generalized metrics computed with continuous $\hat{v}_n$. (Top) Threshold τ_A for "Melody-A" (Bottom) Threshold τ_B for "Melody-B".

We use three melody extraction datasets in our experiments: iKala [10], MedleyDB [5] and Orchset [8]. iKala comprises 252 30-second excerpts sampled from 206 songs. MedleyDB contains 108 melody annotated files, which are mostly full-length songs between 3 and 5 minutes long, and cover a variety of instrumentation and genres. For our experiments, we use the melody 2 definition: the f_0 curve of the predominant melodic line drawn from multiple sources. Finally, Orchset contains 64 short audio excerpts (between 10 and 30 s.) of symphonic music. For pitch tracking, we use a dataset derived from MedleyDB, with 103 tracks of solo, monophonic instruments. In the two datasets for which we have isolated sources readily available, MedleyDB-Pitch and iKala, we compute the reference reward using the method described in Section 5 using a hop size of 256 and a window size of 4096 for a sample rate of 44100 Hz.

6.1 Voicing Estimation Metrics

We first examine the difference in the generalized versus the classic metrics for VR and VFA. In the classic metrics, the choice of voicing threshold has a major effect on VR and VFA. Figure 4 shows the classic metrics as a function of the voicing threshold as dots for VR (red) and VFA (blue) for "Melody-A" and "Melody-B" on the three melody datasets. The dashed horizontal lines show the value of the generalized metric, which is computed independently of the threshold. We see that the value of the generalized metrics is close to the average value of the metrics for all thresholds.

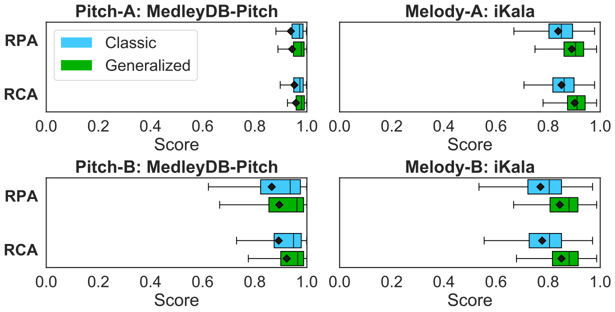

Figure 5. Generalized vs Classic RPA and RCA for four algorithms. r_n is used for the reference datasets. Boxplots show statistics across tracks for each metric.

6.2 Pitch Accuracy Metrics

Figure 5 shows the generalized and classic `RPA` and `RCA` on iKala (for "Melody-A" and "Melody-B") and MedleyDB-Pitch (for "Pitch-A" and "Pitch-B"). We can see that the values of the generalized metrics are higher in all cases, which confirms that algorithms commonly make more errors when the reference reward is lower (more difficult cases). The difference between the classic and generalized metrics is larger on "Melody-B" (+0.08 on average for `RPA`) than "Melody-A" (+0.05 on average for `RPA`), which suggests that "Melody-A" is less prone to pitch estimation errors when the melody is less predominant.

6.3 Overall Accuracy

Finally, we compare the generalized metrics with the classic metrics for `OA`. It is most often used as a single measure to compare the performance of two algorithms, but in the classic metric, the choice of each algorithm's voicing threshold can change the relative ranking of `OA`. In Figure 6, the middle and right columns show the classic `OA` as a function of threshold for two algorithms, and the left column shows the relative ranking of the classic `OA` for each combination of thresholds; when a cell is red, the algorithm in the middle gets a higher value for this metric, and vise versa when a cell is blue. We see that for all datasets, a pair of threshold values can be chosen which rank one algorithm higher than the other. This makes the comparison of two algorithms in terms of the classic `OA` highly dependent on the choice of threshold.

We see that when algorithms are ranked based on the generalized `OA`, the algorithm which is ranked higher is always the algorithm which is also more often ranked higher for the classic `OA` (i.e. the dominant color in Figure 6, left). The ordering of the generalized `OA` is also consistent with the highest possible value of the classic `OA` (the "star" marker in Figure 6). This suggests that the generalized `OA` provides a threshold independent way to fairly rank algorithms. Comparing the generalized metrics with and without r_n (Figure 6, horizontal dashed and solid lines), we see that overall the behavior of `OA` is similar, but harsher when r_n is not used.

7. CONCLUSIONS

This paper presents a generalization of traditional single-f_0 estimation metrics, which allows estimators to provide a continuous voicing estimate and introduces a weighting on pitch accuracy. We perform an experimental comparison of the proposed metrics using both monophonic pitch estimators and melody extraction algorithms and show that the generalized metrics provide a threshold-independent way of comparing algorithms. Additionally, we propose a methodology for the annotation of the reference reward r_n based on the energy of the isolated sources and also propose a promising variant for the case when only polyphonic mixtures are available, based on pitch-informed source separation. One of the limitations of the proposed method for estimating the reference reward is that it does not explicitly consider aspects related to pitch perception, which we leave for future work. Finally, the proposed evaluation framework could also be extended to multiple pitch estimation metrics. The concept of confidence and reward, in this case, would be related to each of the individual pitches present, and the methodology would still hold.

While this paper focuses on the generalization of the classic metrics, we also foresee the creation of new metrics, including the adaptation of metrics from the Information Retrieval literature (such as the ROC-AUC score). The current work only considers voicing confidence for estimators and a kind of "pitch confidence" for references, however pitch confidence for estimators and voicing confidence for references could also be incorporated. Future work could also experiment with metrics based on different types of rewards r_n, such as metrics that examine pitch accuracy in difficult frames.

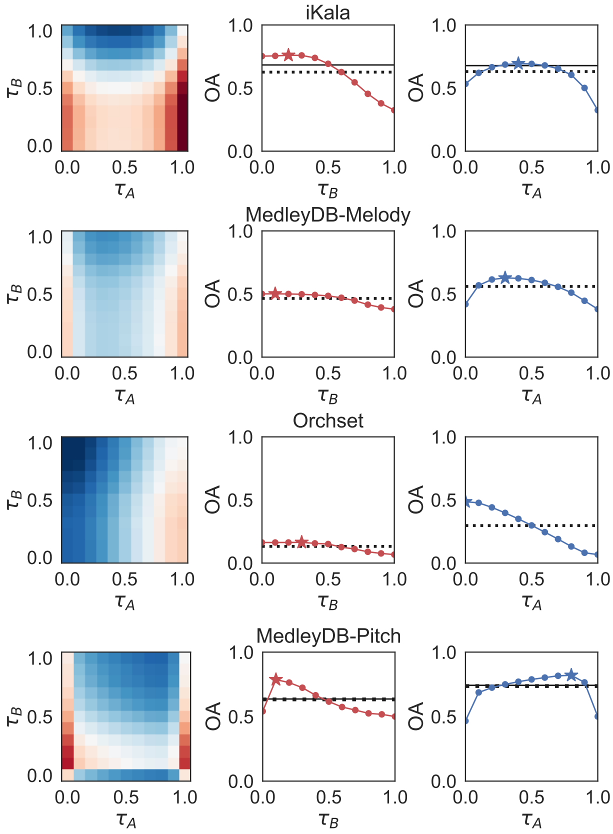

Figure 6. (Left column) difference in overall accuracy between two algorithms. Red: algorithm B gets a higher `OA`, Blue: algorithm A gets a higher `OA`, White: They are the same. (Middle and Right Columns) `OA` as a function of threshold for "Melody-B" (middle) and "Melody-A" (right) for rows 1-3, and for "Pitch-B" (middle) and "Pitch-A" (right) in row 4. Dashed lines show the generalized `OA` computed with $r_n = v_n$ and continuous $\hat{v}_n$, solid lines show the generalized `OA` computed with continuous r_n and continuous $\hat{v}_n$. Solid lines are missing for two datasets because the isolated melody sources are not available so we cannot accurately compute continuous r_n.

8. REFERENCES

[1] V. Arora and L. Behera. On-line melody extraction from polyphonic audio using harmonic cluster tracking. *IEEE Transactions on Audio, Speech, and Language Processing*, 21(3):520–530, March 2013.

[2] D. Basaran, S. Essid, and G. Peeters. Main melody extraction with source-filter nmf and crnn. In *Proceedings of the International Society for Music Information Retrieval Conference (ISMIR)*, 2018.

[3] R. Bittner, B. McFee, J. Salamon, P. Li, and J.P. Bello. Deep salience representations for f0 estimation in polyphonic music. In *Proceedings of the International Society for Music Information Retrieval Conference (ISMIR)*, October 2017.

[4] R. Bittner, J. Salamon, S. Essid, and J. Bello. Melody extraction by contour classification. In *Proceedings of the International Society for Music Information Retrieval Conference (ISMIR)*, pages 500–506, Oct. 2015.

[5] R. Bittner, J. Salamon, M. Tierney, M. Mauch, C. Cannam, and J. Bello. Medleydb: a multitrack dataset for annotation-intensive mir research. In *Proceedings of the International Society for Music Information Retrieval Conference (ISMIR)*, pages 155–160, Oct. 2014.

[6] J. Bosch. *From heuristics-based to data-driven audio melody extraction*. PhD thesis, Universitat Pompeu Fabra, June 2017.

[7] J. Bosch, R. M. Bittner, J. Salamon, and E. Gómez. A comparison of melody extraction methods based on source-filter modelling. In *Proceedings of the International Society for Music Information Retrieval Conference (ISMIR)*, pages 571–577, Aug. 2016.

[8] J. Bosch, R. Marxer, and E. Gómez. Evaluation and combination of pitch estimation methods for melody extraction in symphonic classical music. *Journal of New Music Research*, 45(2):101–117, 2016.

[9] A. Bregman. *Auditory scene analysis: The perceptual organization of sound*. MIT press, 1994.

[10] T. Chan, T. Yeh, Z. Fan, H. Chen, L. Su, Y. Yang, and R. Jang. Vocal activity informed singing voice separation with the ikala dataset. In *2015 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, pages 718–722. IEEE, 2015.

[11] H. Cheveigné, A.and Kawahara. Comparative evaluation of f0 estimation algorithms. In *Seventh European Conference on Speech Communication and Technology*, 2001.

[12] JM. Doughty and WR. Garner. Pitch characteristics of short tones. ii. pitch as a function of tonal duration. *Journal of Experimental Psychology*, 38(4):478, 1948.

[13] K. Dressler. Towards Computational Auditory Scene Analysis: Melody Extraction from Polyphonic Music. In *Proceedings of the International Symposium on Computer Music Multidisciplinary Research (CMMR)*, pages 319–334, 2012.

[14] J. Durrieu, G. Richard, B. David, and C. Févotte. Source/filter model for unsupervised main melody extraction from polyphonic audio signals. *IEEE Transactions on Audio, Speech, and Language Processing*, 18(3):564–575, 2010.

[15] B. Fuentes, A. Liutkus, R. Badeau, and G. Richard. Probabilistic model for main melody extraction using constant-Q transform. In *Proceedings of the IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, pages 5357–5360. IEEE, 2012.

[16] J. Kim, J. Salamon, P. Li, and J.P. Bello. Crepe: A convolutional representation for pitch estimation. In *2018 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, pages 161–165. IEEE, 2018.

[17] S. Kum and J. Nam. Joint detection and classification of singing voice melody using convolutional recurrent neural networks. *Applied Sciences*, 9(7):1324, 2019.

[18] JCR Licklider. Influence of phase coherence upon the pitch of complex, periodic sounds. *The Journal of the Acoustical Society of America*, 27(5):996–996, 1955.

[19] M. Mauch and S. Dixon. Pyin: A fundamental frequency estimator using probabilistic threshold distributions. In *Proceedings of the IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, pages 659–663, May 2014.

[20] B. Nieto. Addressing user satisfaction in melody extraction. Master's thesis, Universitat Pompeu Fabra, 2014.

[21] C.J. Plack, A.J. Oxenham, and R. Fay. *Pitch: neural coding and perception*, volume 24. Springer Science & Business Media, 2006.

[22] G. Poliner, D. Ellis, A. Ehmann, E. Gómez, S. Streich, and B. Ong. Melody transcription from music audio: Approaches and evaluation. *IEEE Transactions on Audio, Speech, and Language Processing*, 15(4):1247–1256, 2007.

[23] F. Rigaud and M. Radenen. Singing voice melody transcription using deep neural networks. In *Proceedings of the International Society for Music Information Retrieval Conference (ISMIR)*, pages 737–743, 2016.

[24] J. Salamon and E. Gómez. Melody extraction from polyphonic music signals using pitch contour characteristics. *IEEE Transactions on Audio. Speech. Lang. Processing*, 20(6):1759–1770, 2012.

[25] J. Salamon, E. Gómez, and J. Bonada. Sinusoid extraction and salience function design for predominant melody estimation. In *Proceedings of the International Conference on Digital Audio Effects (DAFx)*, pages 73–80, 2011.

[26] J. Salamon, E. Gómez, D. Ellis, and G. Richard. Melody extraction from polyphonic music signals: Approaches, applications, and challenges. *IEEE Signal Processing Magazine*, 31(2):118–134, 2014.

[27] P. Singh. Perceptual organization of complex-tone sequences: A tradeoff between pitch and timbre? *The Journal of the Acoustical Society of America*, 82(3):886–899, 1987.

[28] P. Verma and RW. Schafer. Frequency estimation from waveforms using multi-layered neural networks. In *Proceedings of Interspeech*, pages 2165–2169, 2016.