AN ANALYSIS/SYNTHESIS FRAMEWORK FOR AUTOMATIC F0 ANNOTATION OF MULTITRACK DATASETS

Justin Salamon[1*], Rachel M. Bittner[1], Jordi Bonada[2], Juan J. Bosch[2], Emilia Gómez[2] and Juan Pablo Bello[1]

[1]Music and Audio Research Laboratory, New York University, USA

[2]Music Technology Group, Universitat Pompeu Fabra, Spain

*Please direct correspondence to: justin.salamon@nyu.edu

ABSTRACT

Generating continuous f_0 annotations for tasks such as melody extraction and multiple f_0 estimation typically involves running a monophonic pitch tracker on each track of a multitrack recording and manually correcting any estimation errors. This process is labor intensive and time consuming, and consequently existing annotated datasets are very limited in size. In this paper we propose a framework for automatically generating continuous f_0 annotations without requiring manual refinement: the estimate of a pitch tracker is used to drive an analysis/synthesis pipeline which produces a synthesized version of the track. Any estimation errors are now reflected in the synthesized audio, meaning the tracker's output represents an accurate annotation. Analysis is performed using a wide-band harmonic sinusoidal modeling algorithm which estimates the frequency, amplitude and phase of every harmonic, meaning the synthesized track closely resembles the original in terms of timbre and dynamics. Finally the synthesized track is automatically mixed back into the multitrack. The framework can be used to annotate multitrack datasets for training learning-based algorithms. Furthermore, we show that algorithms evaluated on the automatically generated/annotated mixes produce results that are statistically indistinguishable from those they produce on the original, manually annotated, mixes. We release a software library implementing the proposed framework, along with new datasets for melody, bass and multiple f_0 estimation.

1. INTRODUCTION

Research on Music Information Retrieval (MIR) tasks such as melody extraction and multiple f_0 estimation requires audio datasets annotated with precise, continuous, sometimes multiple, f_0 values at time-scales on the order of milliseconds. Generating such annotations manually is very time consuming and labor intensive, and thus insufficient to sustain current research efforts. This is aggravated by the lack of educational or other intrinsic motivations for performing f_0 annotations, limiting the applicability of gamification and other crowdsourcing strategies to this problem. Alternative solutions for f_0 annotation include the use of instruments outfitted with sensors that are able to simultaneously generate audio and annotations [18], or of MIDI-controlled instruments to support annotation by playing [39]. Such approaches are limited either in the type of sources that they can use, e.g. piano, or in the annotations they can generate, e.g. notes instead of continuous f_0. Other approaches rely on audio to MIDI alignment [19], but are limited both by the robustness of the alignment and, to a lesser extent, the availability of good quality MIDI data. Perhaps the most common methodology for annotating f_0 is to use automatic f_0 estimation methods on monophonic stems of existing multitracks [7, 16, 29]. However, the limited accuracy of the estimation has the potential to create discrepancies between the audio and the annotation [16], and correcting such discrepancies is in itself very laborious. For example, manual corrections for MedleyDB (108 songs, most 3–5 minutes long) required approximately 50 hours of effort across annotators [7,29]. As a result, existing datasets for f_0 estimation in polyphonic music (whether for melody, bass, or multiple f_0) are extremely small: most such datasets are on the order of tens of recordings with a total duration of less than an hour. Even MedleyDB is but a fraction of the size of datasets used in other MIR tasks [4], speech recognition [12] or image recognition [14]. This is particularly problematic for developing data-driven solutions to f_0 estimation, which require large amounts of annotated audio data.

To tackle this problem, the MIR community, and the machine learning (ML) community in general, have proposed solutions based on *data augmentation* and *data synthesis*. Augmentation involves the transformation of existing data, and has been shown to improve the generalizability of ML models across domains [25, 31]. However, if the initial dataset is very small there is a limit to the benefits of augmentation, and thus researchers have also explored data synthesis approaches, e.g. for chord recognition [27], monophonic pitch tracking [30] or environmental sound analysis [26]. The earliest dataset for melody extraction, ADC2004 [10], contains some synthesized vo-

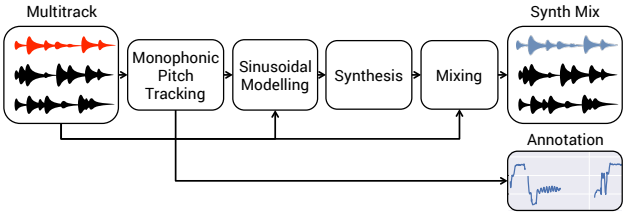

Figure 1. Block diagram of the proposed framework.

cal tracks and is still in use for melody extraction evaluation in MIREX [15] today. Synthesized data is not only useful for model training, it can also be used for model evaluation [26]. As the authors of that study note, while evaluation on synthesized data might not always represent model performance on real-world data, it allows for a detailed and controlled comparative evaluation using significantly larger amounts of data, which can provide invaluable insight into the comparative performance of different models under different, controlled, audio conditions.

Building on these ideas, in this paper we present a method for continuous f_0 annotation that is fully automatic. The key concept is the use of multitrack recordings in combination with an analysis/synthesis framework: starting with a multitrack recording, we select a monophonic instrument track that we are interested in annotating, and run a monophonic pitch tracker to obtain its f_0 curve. Since the f_0 estimate is likely to contain (albeit a small amount of) errors, it would be methodologically unsound to treat it as a reference annotation for either training or evaluation. Instead, we use it as the input to a wide-band harmonic modelling algorithm that estimates not just the frequency of the f_0, but the frequency, amplitude and phase of every harmonic in the signal. We use this information to re-synthesize the monophonic recording, resulting in an audio signal that perfectly matches the f_0 curve produced by the pitch tracker. Thanks to the wide-band harmonic modelling, the synthesized track is very similar to the original recording in pitch, timbre and dynamics [1]. Finally, we mix the synthesized track back with the rest of the instruments in the multitrack recording, resulting in a polyphonic music mixture for which we have an accurate, fully automatic annotation of the synthesized track. A block diagram of the proposed framework is displayed in Figure 1. The methodology can be used to automatically generate annotations for working on melody extraction, bass line extraction and multiple f_0 estimation, and essentially any model designed to extract f_0 content from polyphonic music mixtures.

The proposed framework can be readily used to generate training data. The question remains whether using the synthesized mixes as evaluation data produces a representative measure of model performance. To answer this question, after describing the framework we present a series of experiments designed to explore whether the synthesized mixes result in performance scores that are representative of the scores algorithms obtain on the original mixes. As a final contribution of this work, we release a software library implementing the proposed framework [2], as well as new datasets for melody, bass, and multiple f_0 estimation [3].

[1] For examples of synthesized tracks (solo and mixed with the multitrack) see: `http://synthdatasets.weebly.com/examples`

2. METHOD

2.1 Pitch track analysis/synthesis

2.1.1 Pitch tracking

We use a monophonic pitch tracker to get an initial f_0 estimate of the stem we would like to annotate. We tested SAC [21] and YIN [13] and compared both to the manually corrected f_0 annotations provided in MedleyDB [7]. Based on this comparison we decided to use SAC for our experiments, see Section 3.2 for further details. The output of SAC is automatically cleaned by filling short gaps (<50 ms), removing short voiced segments (<50 ms), and smoothing the voiced segments. Note that we do not use pYIN [30], a state-of-the-art pitch tracking algorithm, since the manually corrected annotations in MedleyDB are based on the output of this algorithm and so using it for this stage could bias our experimental results. Still, it is important to note that the methodology is independent of the specific pitch tracker used, and the software library we release supports multiple monophonic pitch trackers, including pYIN.

2.1.2 Sinusoidal modelling

We use the wide-band harmonic sinusoidal modelling algorithm [8] for estimating the harmonic parameters (frequency, amplitude and phase) at every signal period. The algorithm first segments the signal into periods corresponding to the fundamental frequency. Then each period is analyzed with a certain windowing configuration that has the property that the Fourier transform of the window has the zeros located at multiples of the f_0. This property reduces the interference between harmonics, and allows the estimation of harmonic parameters using a temporal resolution close to one period of the signal. For details see [8].

2.1.3 Synthesis

The synthesis is performed with a bank of oscillators. The harmonics' parameters previously estimated are linearly interpolated at the synthesis sampling rate. Frequencies are set to exact multiples of the f_0. Phases are arbitrarily initialized at each voiced segment with a non-flat shape to avoid producing signals that are too peaky:

$$\Phi_h = \pi + \frac{\pi}{2} \sin\left(\frac{h}{20\pi} + \pi\right) \quad (1)$$

where h corresponds to the harmonic index, and Φ is the harmonic phase. Phases are incremented at each sample using the interpolated frequency value. At voiced segment boundaries harmonic amplitudes are faded out to zero within one signal period. Unvoiced segments are muted.

[2] `https://github.com/marl/massage`

[3] `http://synthdatasets.weebly.com/`

2.2 Remixing

The final step is to recreate a mix of the song that is as close as possible to the original. Even when using the original stems as source material, a simple unweighted sum of the stems will not necessarily be a good approximation: the stems may not be the same volume as they occur in the mix, and the final mix may have mastering effects such as compression or equalization. To estimate the mixing weights, we model the (time-domain) mix $y[n]$ as a weighted linear combination [4] of the original stems $x_1, x_2, \ldots, x_M$:

$$y[n] \approx \sum_{i=1}^{M} a_i x_i[n] \tag{2}$$

where $x_i[n]$ is the audio signal at sample n for stem i and M is the total number of stems. Let N be the total number of samples in each audio signal. We then estimate the mixing weights a_i by minimizing a non-negative least squares objective $||X\mathbf{a} - Y||_2$ over $\mathbf{a}$ for $a_i > 0$, where X is the $N \times M$ matrix of the absolute values of the stem audio signals $|x[n]|$, $\mathbf{a}$ is the $M \times 1$ vector of mixing weights a_i, and Y the $N \times 1$ is the absolute value of the mixture audio signal $|y[n]|$. We use the computed weights $\mathbf{a}$ to create a (linear) remix $\tilde{y}[n]$, substituting the melody track(s) (or bass track or multiple instrument tracks) $\tilde{x}_1, \ldots, \tilde{x}_I$ with the synthesized stems:

$$\tilde{y}[n] = \sum_{i=1}^{I} a_i \tilde{x}_i[n] + \sum_{i=I+1}^{M} a_i x_i[n] \tag{3}$$

3. EXPERIMENTS

As noted above, the proposed framework can be readily used for generating training data. However, and perhaps precisely due to the problem of data scarcity, current state-of-the-art algorithms for melody extraction (e.g., [9, 17, 35]) and multiple f_0 estimation (e.g., [3, 16, 24]) are either fully or partially based on heuristic DSP pipelines, meaning it is not possible to demonstrate an improvement due to additional training data, as these systems do not have a learning stage (or the learning happens towards the end of the pipeline and the main source of errors is the heuristic front-end [6]). We are actively working on f_0 estimation algorithms based on deep models that operate on a low-level representation of the signal [5], and plan to evaluate their performance when trained on synthesized data as part of our future work.

Instead, we explore the representativeness of the synthesized mixes for the purpose of model evaluation. To this end, we run a series of evaluation experiments, once using the original mixes and annotations and a second time using the synthesized mixes and automatically generated annotations. The experiments involve evaluating several melody extraction and multiple f_0 estimation algorithms. Ideally, we would like the scores obtained by each algorithm to remain unchanged between the original and synthesized mixes, as this would indicate that the synthesized (automatically annotated) mixes can be used to obtain realistic estimates of model performance, opening the door to the generation of significantly larger datasets not only for model training, but also for model evaluation.

3.1 Data

We use the MedleyDB dataset [7] to evaluate the proposed methodology for melody f_0 annotation. Of the 108 tracks containing melodies, we need to filter out tracks that are not completely monophonic such as those containing recording bleed from other instruments and melody tracks played by polyphonic instruments such as the piano and guitar. After filtering we end up with 65 songs, for which we generate new mixes and melody f_0 annotations following the methodology described in Section 2. The remixing is performed using the `medleydb` python module [5]. We call the resulting dataset MDB-melody-synth.

For multiple f_0 estimation we use the Bach10 dataset [16]. The dataset contains ten pieces of four-part (soprano, alto, tenor, bass) J.S. Bach chorales performed by the violin, clarinet, saxophone and bassoon, respectively. The synthesized dataset including new mixes and automatically generated multiple f_0 annotations, Bach10-mf0-synth, was created following the methodology described in Section 2, the only difference being that since the original mixes are just unweighted sums of the stems, the synthesized mixes are also unweighted.

Finally, we use the proposed methodology to create a synthesized version of MedleyDB with multiple f_0 annotations, MDB-mf0-synth, and another version in which only the bass track is synthesized (for bass line extraction), MDB-bass-synth. For MDB-mf0-synth, we need to filter out stems that are not monophonic. For instance, if the original mix contains drums, bass, piano, guitar, trumpet and singing voice, the new mix will contain drums, bass, trumpet and voice. We must also discard tracks that are left with only percussive instruments after removing all non-monophonic stems. After filtering we are left with 85 songs, for which we generate new mixes and multiple f_0 annotations as per Section 2. Most of the mixes in the resulting dataset have a polyphony between 1 and 4, but there are also songs with higher polyphonies, up to 16. Overall, the mixes in the new dataset include 25 different instruments (not counting percussive instruments) which are combined to produce 29 unique instrumentations (not counting percussive instruments). For MDB-bass-synth we can use all tracks that contain a bass line with no recording bleed, resulting in a dataset of 71 songs. To the best of our knowledge this is the largest publicly available dataset with continuous bass f_0 annotations. Note that due to space constraints we do not use this dataset in the experiments reported in this paper. All four new datasets, MDB-melody-synth, MDB-mf0-synth, MDB-bass-synth and Bach10-mf0-synth are made freely available online (cf. footnote 3).

[4] Recreating mastering effects is left for future work.

[5] `https://github.com/marl/medleydb`

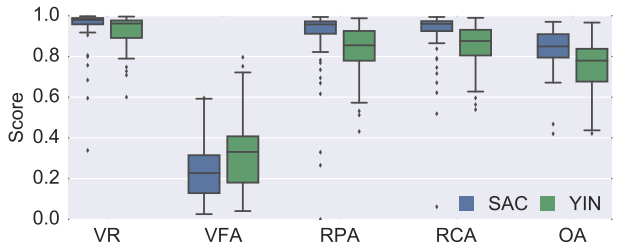

Figure 2. f_0 tracking scores for SAC and YIN evaluated against the MedleyDB manually corrected f_0 annotations.

3.2 Monophonic Pitch Tracking

We start by evaluating the pitch tracking accuracy of the SAC and YIN algorithms on the 65 monophonic melody stems from MedleyDB, presented in Figure 2. We use `mir_eval` [33] to compute the standard five evaluation metrics used in MIREX: Voicing Recall (VR), Voicing False Alarm (VFA), Raw Pitch Accuracy (RPA), Raw Chroma Accuracy (RCA) and Overall Accuracy (OA). For details about the metrics see [36]. We see that SAC produces a more accurate f_0 estimate compared to YIN for these data, with a mean raw pitch accuracy of 0.9. The overall accuracy is slightly lower due to voicing false positives, but these frames will turn into voiced frames in the synthesized mixes thus accurately matching the annotation. This is the key advantage of the proposed approach: pitch tracking errors do not cause a mismatch between the audio and the annotation and require no manual correction. Since 90% of the f_0 values in MDB-melody-synth match those in MedleyDB, we can also safely say the synthesized dataset is representative of the original in terms of continuous pitch values. Finally, since SAC makes practically no octave errors (the difference between the RPA and RCA is below 0.02), there is little to no risk of a perceptual mismatch between the estimated f_0 and the synthesized audio.

3.3 Melody extraction

To evaluate the representativeness of MDB-melody-synth compared to MedleyDB, we evaluate the performance of three melody extraction algorithms: Melodia [35], the source-separation-based algorithm by Durrieu [17], and the recently proposed algorithm by Bosch [9] which uses a salience function based on Durrieu's model in combination with the contour characterization employed in Melodia for voicing detection and melody selection.

In Figure 3(a) we plot the results obtained by the Melodia algorithm, where for each metric we plot the result for the original mixes and the MDB-melody-synth mixes side-by-side. We see that while the results are not identical, the distribution of scores for each metric remains stable. A two-sided Kolmogorov-Smirnov test confirms that for all 5 metrics the differences in the score distributions between the original and synthesized datasets are not statistically significant (p-values of 0.39, 0.05, 0.68, 0.28 and 0.82 for VR, VFA, RPA, RCA and OA respectively). We repeat the same experiment for the algorithms by Durrieu and Bosch, displayed in Figure 3 subplots (b) and (c) respectively. As before, the score distributions for all metrics remain stable and the difference between them is not statistically significant. The only exception is the OA score for Durrieu's algorithm: this is an artefact of the algorithm's tendency to report most frames as voiced, which leads to a small increase in OA given that MDB-melody-synth contains slightly more voiced frames compared to MedleyDB. Still, reporting most frames as voiced also heavily penalizes the algorithm (on both datasets), and despite the increase in OA the algorithm remains consistently ranked below Melodia and Bosch's algorithm in terms of OA. Indeed, the relative *ranking* of all three algorithms in terms of pitch and overall accuracy remains unchanged between MedleyDB and MDB-melody-synth, as shown in Figure 4.

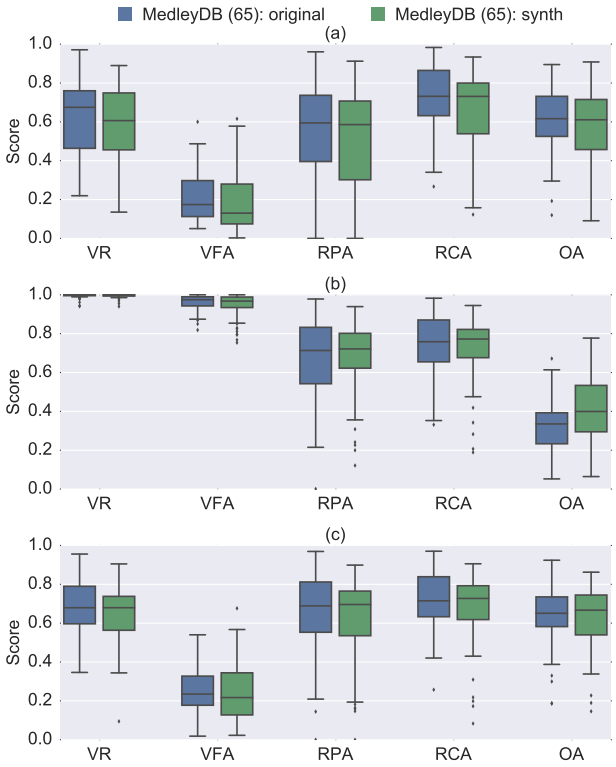

Figure 3. Melody extraction evaluation scores for 65 songs: (blue) original MedleyDB mixes and (green) MDB-melody-synth mixes. (a) Melodia, (b) Durrieu, (c) Bosch.

3.4 Multiple f_0 estimation

As noted earlier, we use the Bach10 dataset [16] to evaluate the representativeness of the synthesized mixes resulting from our proposed methodology for multiple f_0 estimation. For this task 14 different metrics are computed in MIREX. It suffices to know that the first six measure "goodness" and go from 0 (worst) to 1 (best): Precision, Recall, Accuracy, and a chroma version (ignoring octave errors) for each, which we indicate with a "C_" prefix in our plots. The latter eight measure four different types of errors and their chroma counterparts, where 0 is the best score and greater values mean more errors. The reader is referred to [2, 32] for a detailed description of each metric. As before, all metrics are computed with `mir_eval`.

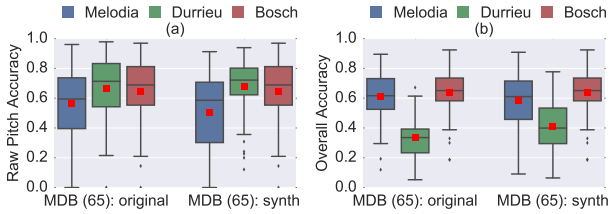

Figure 4. Evaluation scores for the three melody extraction algorithms on 65 MedleyDB and MDB-melody-synth mixes: (a) Raw Pitch Accuracy and (b) Overall Accuracy.

We use two multiple f_0 estimation algorithms for our evaluation: those by Benetos [3] and Duan [16]. The results are presented in Figure 5. For Benetos's method there is no statistically significant difference between Bach10 and Bach10-mf0-synth for any of the 14 metrics, and for Duan's there is no statistically significant difference for 10 of the 14 including the most important metrics such as Recall, Precision, Accuracy, and E_tot. The relative ranking of the two algorithms remains unchanged for all 14 metrics, as shown in Figure 6 subplots (a), (b), and (c) for Precision, Recall, and Accuracy respectively.

Since MedleyDB does not include multiple f_0 annotations, we cannot compare the performance of Benetos's and Duan's algorithms on MDB-mf0-synth to the original dataset as we did for MDB-melody-synth and Bach10-mf0-synth. In essence, MDB-mf0-synth is a completely new dataset for evaluating multiple f_0 estimation algorithms. The results obtained by Benetos's and Duan's algorithms for this new dataset are presented in Figure 7. We see that the performance of both algorithms drops considerably compared to the results they obtain on Bach10 (note the change in y-axis range), indicating that this new dataset is more challenging. The difference in performance between the two algorithms is smaller, and both seem to make an increased number of octave errors compared to Bach10, as indicated by the greater difference between the metrics and their chroma counterparts. The false alarm rate (E_fa) for both algorithms is also greater, which could be due to the greater proportion of tracks in MDB-mf0-synth with low polyphonies compared to Bach10, or due to the presence of percussive sources which are completely absent from the latter. Another interesting result is the significantly higher *variance* of all the metrics on MDB-mf0-synth compared to Bach10, which is likely due to the considerably greater variety in MDB-mf0-synth in terms of musical genre, instrumentation and polyphony. As an example of the performance analysis that can be done using MDB-mf0-synth, in Figure 8 we present the accuracy scores for the two algorithms broken down by polyphony. While it is beyond the scope of this paper, similar breakdowns could be performed by genre, instrumentation, vocal/instrumental, the presence/absence of percussion, etc.

4. DISCUSSION

We have proposed a methodology for the automatic f_0 annotation of polyphonic music by means of multitrack

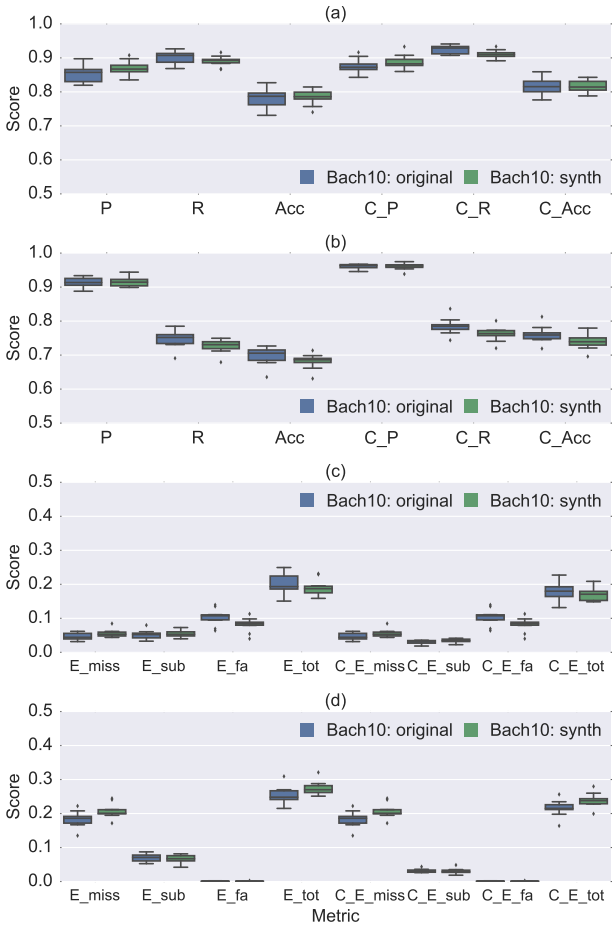

Figure 5. Multiple f_0 estimation scores on the Bach10 dataset, original mixes (blue) and synthesized mixes (green): (a) Benetos (b) Duan (c) Benetos errors (d) Duan errors. The chroma versions of each metric are indicated by a "C_" prefix.

datasets and an analysis/synthesis framework. We applied this methodology to create automatic f_0 annotations for melody extraction, bass line extraction and multiple f_0 estimation using the MedleyDB and Bach10 datasets. As noted in the introduction, these datasets can be used to train learning based f_0 estimation algorithms, as well as conduct controlled evaluation experiments. Furthermore, by means of a comparative evaluation we have shown that algorithms evaluated against the synthesized mixes and automatically generated f_0 annotations produce results that are, in almost all cases, equivalent (up to statistical significance) to those they produce for the original mixes. This suggests that in addition to providing insight from large-scale evaluation and facilitating multiple controlled evaluation breakdowns, the results are in fact quite representative (in terms of absolute scores) of the results we would have obtained by manually annotating the original mixes.

Since the proposed methodology is scalable and fully automatic, it can be readily applied to other existing multitrack datasets [1, 20, 22, 28, 37, 41], most of which were originally intended for source separation or automatic mixing evaluation. It can also be applied to datasets that provide separate melody and accompaniment tracks [11, 23].

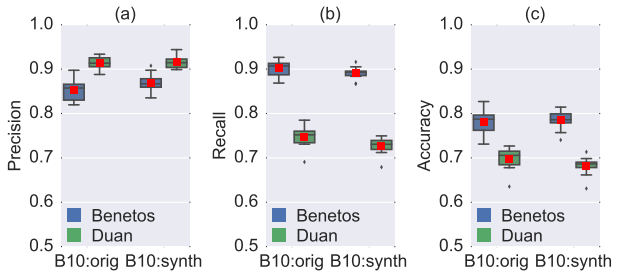

Figure 6. Multiple f_0 estimation scores for Duan's and Benetos's algorithms on Bach10 (B10:orig) and Bach10-mf0-synth: (a) Precision, (b) Recall and (c) Accuracy.

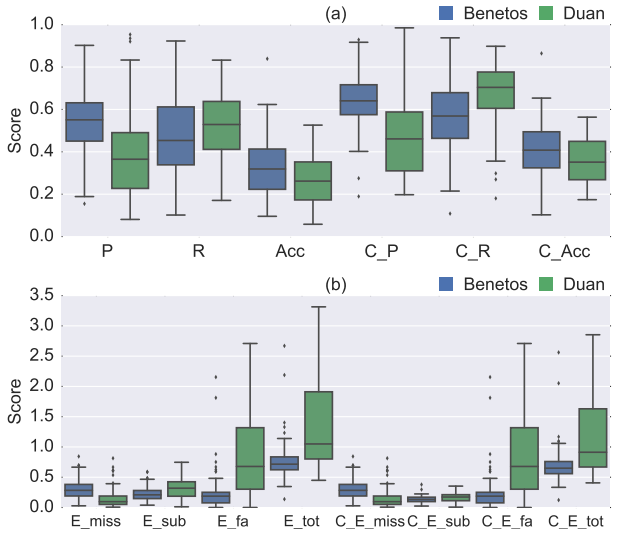

Figure 7. Evaluation scores for the multiple f_0 estimation algorithms by Benetos and Duan on the new MDB-mf0-synth dataset: (a) score metrics, (b) error metrics.

An important limitation of our methodology is that it can only be applied to monophonic stems, meaning it cannot be used to annotate polyphonic instruments such as the piano and the guitar. To address this, we are currently working on expanding the proposed framework by incorporating polyphonic transcription algorithms that can be applied in place of the monophonic pitch tracker for executing the first stage of the proposed framework on polyphonic stems. It can also be argued that since our approach requires generating new mixes (with a subset of the tracks replaced by synthesized versions), the resulting audio data do not reflect real-world data as reliably as the original mixes. While this is true, the results of our experiments suggest that the scores obtained using the synthesized datasets are in fact to a great extent representative of those one would obtain on the original mixes. Furthermore, since existing datasets for f_0 estimation in polyphonic music are so small, it is unlikely for the results obtained on these datasets to generalize to significantly larger audio collections, regardless of how they were annotated. We believe that the benefits of training and evaluating f_0 estimation algorithms on large-scale datasets with significantly greater variety in terms of audio content, enabled by our proposed framework, outweigh its limitations and have

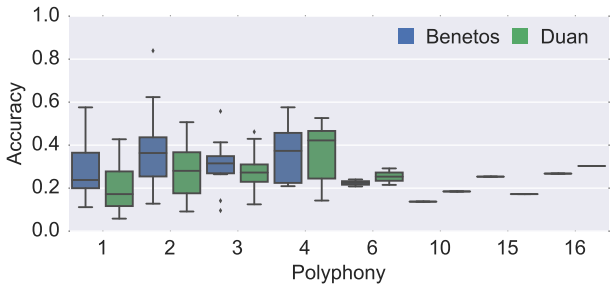

Figure 8. Accuracy scores for the algorithms by Benetos and Duan on MDB-mf0-synth, by polyphony.

the potential to lead to new insights and novel models for f_0 estimation in polyphonic music.

As research on analysis/synthesis algorithms and automatic mixing [34, 37, 38] advances, we can expect our framework to produce mixes that are increasingly authentic and true to the original mixes. The synthesis used in this study is purely harmonic, which affects the quality of the synthesis and could potentially affect the perception of note onsets (e.g., vocals with fricatives). We are currently expanding the framework to support harmonic+noise synthesis, and updated versions of the released datasets will be made available on the companion website. Still, it is important to highlight that the key contribution of this work is the proposed methodology itself, and our experimental results showing the representativeness of the mixes and annotations it produces. The value of this framework is precisely in the fact that we can use analysis and synthesis algorithms which, despite not being perfect, produce data of sufficient quality to be of value for MIR research. It means we can generate datasets whose size is only constrained by our (ever growing) access to multitrack recordings.

In a recent study [39], Su and Yang define four criteria for assessing the "goodness" of a dataset and its annotations for evaluating automatic music transcription (AMT) algorithms, which we summarize here: (1) Generality: the form, genre and instrumentation of the music excerpts should be representative of the music universe to which we expect the algorithm to generalize [6] ; (2) Efficiency: the annotation process should be fast and scalable; (3) Cost: the cost of building the dataset, in terms of money and human resources, should be minimized. (4) Quality: the annotations should be accurate enough to facilitate correct evaluation of AMT algorithms. The methodology proposed in this paper satisfies these criteria to a great extent: since the generation of annotations only depends on the availability of multitrack data, it is relatively independent of (1) and can be applied to most musical genres. With regards to criteria (2), (3), and (4): since our methodology generates annotations completely automatically, one could argue that it is as efficient as any annotation technique could possibly be. For the same reason, it is also very cost efficient, since creating annotations is essentially free. Finally, the quality of the annotations is guaranteed since the synthesized tracks match the annotations perfectly.

[6] For a detailed discussion of these considerations see [40].

5. REFERENCES

[1] J. Abeßer, O. Lartillot, C. Dittmar, T. Eerola, and G. Schuller. Modeling musical attributes to characterize ensemble recordings using rhythmic audio features. In *IEEE ICASSP*, pages 189–192, May. 2011.

[2] M. Bay, A. Ehmann, and J. S. Downie. Evaluation of multiple-F0 estimation and tracking systems. In *10th Int. Soc. for Music Info. Retrieval Conf.*, pages 315–320, Kobe, Japan, Oct. 2009.

[3] E. Benetos, S. Cherla, and T. Weyde. An efficient shift-invariant model for polyphonic music transcription. In *6th Int. Workshop on Machine Learning and Music*, pages 1–4, Prague, Czech Republic, Sep. 2013.

[4] T. Bertin-Mahieux, D. P. W. Ellis, B. Whitman, and P. Lamere. The million song dataset. In *12th Int. Soc. for Music Info. Retrieval Conf.*, pages 591–596, Miami, USA, Oct. 2011.

[5] R. M. Bittner, B. McFee, J. Salamon, P. Li, and J. P. Bello. Deep salience representations for f_0 estimation in polyphonic music. In *18th Int. Soc. for Music Info. Retrieval Conf.*, Suzhou, China, Oct. 2017.

[6] R. M. Bittner, J. Salamon, S. Essid, and J. P. Bello. Melody extraction by contour classification. In *16th Int. Soc. for Music Info. Retrieval Conf.*, Malaga, Spain, Oct. 2015.

[7] R. M. Bittner, J. Salamon, M. Tierney, M. Mauch, C. Cannam, and J. P. Bello. MedleyDB: A multitrack dataset for annotation-intensive MIR research. In *15th Int. Soc. for Music Info. Retrieval Conf.*, pages 155–160, Taipei, Taiwan, Oct. 2014.

[8] J. Bonada. Wide-band harmonic sinusoidal modeling. In *11th Int. Conf. on Digital Audio Effects (DAFx-08)*, pages 265–272, Espoo, Finland, Sep. 2008.

[9] J. J. Bosch, R. M. Bittner, J. Salamon, and E. Gómez. A comparison of melody extraction methods based on source-filter modelling. In *17th Int. Soc. for Music Info. Retrieval Conf.*, New York City, USA, Aug. 2016.

[10] P. Cano, E. Gómez, F. Gouyon, P. Herrera, M. Koppenberger, B. Ong, X. Serra, S. Streich, and N. Wack. ISMIR 2004 audio description contest. Technical report, Music Technology Group, Universitat Pompeu Fabra, Barcelona, Spain, Apr. 2006.

[11] T.-S. Chan, T.-C. Yeh, Z.-C. Fan, H.-W. Chen, L. Su, Y.-H. Yang, and R. Jang. Vocal activity informed singing voice separation with the iKala dataset. In *IEEE ICASSP*, pages 718–722, 2015.

[12] S. F. Chen, B. Kingsbury, Lidia Mangu, D. Povey, G. Saon, H. Soltau, and G. Zweig. Advances in speech transcription at IBM under the DARPA EARS program. *IEEE Trans. on Audio, Speech, and Language Processing*, 14(5):1596–1608, Sep. 2006.

[13] A. de Cheveigné and H. Kawahara. YIN, a fundamental frequency estimator for speech and music. *J. Acoust. Soc. Am.*, 111(4):1917–1930, Apr. 2002.

[14] J. Deng, W. Dong, R. Socher, L. J. Li, K. Li, and F.-F. Li. ImageNet: A large-scale hierarchical image database. In *IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 248–255, Miami, FL, USA, Jun. 2009.

[15] J. Stephen Downie. The music information retrieval evaluation exchange (2005–2007): A window into music information retrieval research. *Acoustical Science and Technology*, 29(4):247–255, Jul. 2008.

[16] Z. Duan, B. Pardo, and C. Zhang. Multiple fundamental frequency estimation by modeling spectral peaks and non-peak regions. *IEEE Trans. on Audio, Speech, and Language Processing*, 18(8):2121–2133, 2010.

[17] Jean-Louis Durrieu, Gaël Richard, Bertrand David, and Cédric Févotte. Source/filter model for unsupervised main melody extraction from polyphonic audio signals. *IEEE TASLP*, 18(3):564–575, March 2010.

[18] V. Emiya, R. Badeau, and B. David. Multipitch estimation of piano sounds using a new probabilistic spectral smoothness principle. *IEEE TASLP*, 18(6):1643–1654, 2010.

[19] S. Ewert, M. Müller, and P. Grosche. High resolution audio synchronization using chroma onset features. In *ICASSP*, pages 1869–1872, Taipei, Taiwan, Apr. 2009.

[20] J. Fritsch. High quality musical audio source separation. Master's thesis, UPMC / IRCAM / Telecom ParisTech, 2012.

[21] E. Gómez and J. Bonada. Towards computer-assisted flamenco transcription: An experimental comparison of automatic transcription algorithms from a cappella singing. *Computer Music journal*, 37(2):73–90, 2013.

[22] S. Hargreaves, A. Klapuri, and M. Sandler. Structural segmentation of multitrack audio. *IEEE TASLP*, 20(10):2637–2647, 2012.

[23] C.-L. Hsu and J.-S.R. Jang. On the improvement of singing voice separation for monaural recordings using the MIR-1K dataset. *IEEE Trans. on Audio, Speech, and Language Processing*, 18(2):310–319, Feb. 2010.

[24] A. Klapuri. Multiple fundamental frequency estimation based on harmonicity and spectral smoothness. *IEEE Trans. on Speech and Audio Processing*, 11(6):804–816, Nov. 2003.

[25] A. Krizhevsky, I. Sutskever, and G.E. Hinton. ImageNet classification with deep convolutional neural networks. In *Advances in neural information processing systems (NIPS)*, pages 1097–1105, 2012.

[26] G. Lafay, M. Lagrange, M. Rossignol, E. Benetos, and A. Röbel. A morphological model for simulating acoustic scenes and its application to sound event detection. *IEEE/ACM Trans. on Audio, Speech, and Lang. Proc.*, 24(10):1854–1864, Oct. 2016.

[27] K. Lee and M. Slaney. Acoustic chord transcription and key extraction from audio using key-dependent HMMs trained on synthesized audio. *IEEE Trans. on Audio, Speech, and Language Processing*, 16(2):291–301, Feb. 2008.

[28] A. Liutkus, R. Badeau, and G. Richard. Gaussian processes for underdetermined source separation. *IEEE Trans. on Signal Processing*, 59(7):3155–3167, 2011.

[29] M. Mauch, C. Cannam, R. Bittner, G. Fazekas, J. Salamon, J. Dai, J. P. Bello, and S. Dixon. Computer-aided melody note transcription using the tony software: Accuracy and efficiency. In *TENOR*, Paris, France, 2015.

[30] M. Mauch and S. Dixon. pYIN: A fundamental frequency estimator using probabilistic threshold distributions. In *IEEE Int. Conf. on Acoustics, Speech and Signal Processing (ICASSP)*, pages 659–663, Florence, Italy, May 2014.

[31] B. McFee, E.J. Humphrey, and J.P. Bello. A software framework for musical data augmentation. In *16th Int. Soc. for Music Info. Retrieval Conf.*, pages 248–254, Malaga, Spain, Oct. 2015.

[32] G. E. Poliner and D. P. W. Ellis. A discriminative model for polyphonic piano transcription. *EURASIP Journal on Applied Signal Processing*, 2007(1):154–154, 2007.

[33] 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 *15th ISMIR*, pages 367–372, Taipei, Taiwan, 2014.

[34] J.D. Reiss. Intelligent systems for mixing multichannel audio. In *17th Int. Conf. on Digital Signal Processing*, pages 1–6, Corfu, Greece, Jul. 2011.

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

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

[37] J. Scott and Y. E. Kim. Instrument identification informed multi-track mixing. In *14th Int. Soc. for Music Info. Retrieval Conf.*, pages 305–310, Nov. 2013.

[38] J. Scott, M. Prockup, E. M. Schmidt, and Y. E. Kim. Automatic multi-track mixing using linear dynamical systems. In *8th SMC Conf.*, Jul. 2011.

[39] L. Su and Y.-H. Yang. Escaping from the abyss of manual annotation: New methodology of building polyphonic datasets for automatic music transcription. In *CMMR*, pages 221–233, Plymouth, UK, Jun. 2015.

[40] J. Urbano, M. Schedl, and X. Serra. Evaluation in music information retrieval. *J. of Intelligent Info. Systems*, 41:345–369, 2013.

[41] E. Vincent, S. Araki, F. Theis, G. Nolte, P. Bofill, H. Sawada, A. Ozerov, V. Gowreesunker, D. Lutter, and N. Q. K. Duong. The signal separation evaluation campaign (2007–2010): Achievements and remaining challenges. *Signal Processing*, 92(8):1928–1936, Aug. 2012.