Audio Engineering Society

Conference Paper

Presented at the Conference on
Semantic Audio
2017 June 22 – 24, Erlangen, Germany

Pitch Contours as a Mid-Level Representation for Music Informatics

Rachel M. Bittner[1], Justin Salamon[1,2], Juan J. Bosch[3], and Juan P. Bello[1]

[1]*Music and Audio Research Lab, New York University, USA*
[2]*Center for Urban Science and Progress, New York University, USA*
[3]*Music Technology Group, Universitat Pompeu Fabra, Spain*

Correspondence should be addressed to Rachel Bittner (rachel.bittner@nyu.edu)

ABSTRACT

Content-based Music Informatics includes tasks that involve estimating the pitched content of music, such as the main melody or the bass line. To date, the field lacks a good machine representation that models the human perception of pitch, with each task using specific, tailored representations. This paper proposes factoring pitch estimation problems into two stages, where the output of the first stage for all tasks is a multipitch contour representation. Further, we propose the adoption of *pitch contours* as a unit of pitch organization. We give a review of the existing work on contour extraction and characterization and present experiments that demonstrate the discriminability of pitch contours.

1 Introduction

Content-based Music Informatics includes the extraction of semantically meaningful information from music audio signals. A class of these problems involve estimating information about the pitch content in music, such as melody extraction [1], bass tracking [2], or multiple-f_0 tracking [3]. Figure 1 illustrates the pitch content in an excerpt of pop music, with all the pitches corresponding to a particular instrument highlighted in the same color. Pitch content is often represented in the form of "notes", with a duration, center frequency, and amplitude. However, this representation is incomplete: humans perceive pitch not only as static combinations of harmonically related sinusoids, but also as continuous, time-varying trajectories or streams [4].

Factoring complex problems into a cascade of simpler problems is a well-worn strategy that has been successfully used in both speech and music; factorization allows each component of the system to solve a simpler problem than a single end-to-end model would have to. Typically, problems are broken down into semantically meaningful stages based on domain knowledge. For example, chord recognition models often rely on intermediate pitch representations such as chroma [5] or even guitar fretboard shapes [6]. Cover song identification methods are typically based on chroma, timbre, and melody estimations [7]. Many approaches to melody and bass line estimation use pitch contours as a mid-level representation [1].

In this paper, we present a case for factoring problems involving estimating and characterizing pitch content in musical audio into two stages, where the mid-level

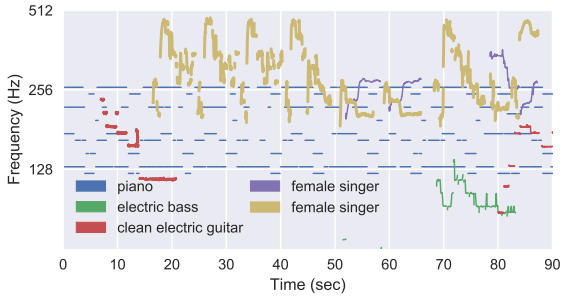

Fig. 1: Pitch contours for every source in an excerpt of pop music. Contours for a single source are drawn in the same color. The main melody line is drawn with a thicker line; from 0 to 15 seconds the main melody is in the clean electric guitar, and after in the (yellow) vocal line.

representation in this factorization are *pitch contours*. We review the strengths and weaknesses in their estimation, and explore what can be achieved with pitch contours estimated from isolated sources for a variety of tasks. We discuss what can be achieved when contours are estimated from polyphonic audio, and finally, propose avenues for future work in pitch estimation tasks. The code used in this paper is available online for reproducibility [1].

2 Motivation

2.1 Why Factored Systems?

With the growing popularity of deep learning architectures, a number of Music Informatics systems have been proposed that perform end-to-end learning to avoid the potentially sub-optimal feature extraction stage. However, it is important to note that deep learning architectures can be applied to "factored" problems too, and they need not solve the problem end-to-end. Furthermore, there is little data available for training pitch estimation tasks [8], making end-to-end systems infeasible, as they require extremely large amounts of labeled data. When large amounts of data are not available, an explicitly-defined mid-level representation can be extremely useful because it constrains the number of free parameters in a system, inherently requiring less data to fit the model. This is especially true when the mid-level representation is known to be meaningful, if not necessary, to the task at hand.

[1] github.com/rabitt/aes-semantic-audio-2017

End-to-end approaches to automatic speech recognition (ASR) are being explored [9], however, these models, while promising, have not yet matched the performance of state-of-the-art factored systems despite being trained on 10,000 hours of labeled speech. The speech community has historically relied on a factored model for automatic speech recognition (ASR), estimating phonemes as an intermediate step in transcribing text from audio [10], and systems built upon this factorization remain state-of-the-art today. Half of the problem—predicting phonemes from audio—is estimated with one type of model, and the second half (predicting sentences from phonemes) is estimated from another. Phoneme recognition is a task that does not require large scale semantic models—phonemes can be reliably estimated from relatively short segments of audio without explicit knowledge of sentence structure or the language being spoken. Once the probabilities for each phoneme are predicted from audio, a second model built to capture sequence structures relies on the phoneme probabilities as input. For example, the phoneme sequence "r-eh-d" can be written "red" or "read"; in a factored model the output of the phoneme transcription portion is the same for both cases, and the language portion need only infer which spelling is correct given the phoneme sequence and the surrounding context. Without factorization, a single model needs to infer the word and its context directly from the audio.

2.2 Why Pitch Contours?

An ideal mid-level representation for pitch estimation tasks should have several characteristics. First and foremost, it must be complete, in that all the pitched content contained in the audio signal is reflected in the representation. For example, it should include the pitches of all instruments in a musical piece as illustrated in Figure 1. Second, unlike notes, it should maintain the variations in pitch and amplitude over time in order to capture musical phenomena such as vibrato, bends or melismas. Finally, it should group pitch content into meaningful semantic units that reflect the grouping a human might create.

A representation that captures all pitched content benefits multiple tasks, including melody extraction, bass tracking, vocal f_0 tracking, multiple-f_0 tracking, music transcription, source separation, and even tasks such as chord identification. In order for such a representation to be useful for multiple tasks, it must be a "multi-pitch" representation, i.e., it must capture the pitch of

all instruments rather than that of a subset or single instrument. One might argue that using a representation that contains more information than is necessary to solve a task can make the subsequent steps of the solution more difficult, however the literature shows that this is not necessarily the case. For example, in melody and bass extraction, methods that have used a mid-level multipitch representation [11, 12, 13, 14] perform well, suggesting that it is not imperative that a mid-level representation be task-specific. Other melody extraction methods create representations designed to extract the pitch content of the melody and to ignore content that is likely not part of the melody [1, 2, 15, 16, 17]. These "melody-oriented" representations have the advantage that the task of selecting the final melody is simplified. However, they can also suffer from having low recall, de-emphasizing some melodic content along with undesired content [18, 19, 20, 21, 22], which imposes an artificial upper bound on performance. Additionally, melody-oriented representations are not useful for other tasks, such as multiple-f_0 or bass tracking, necessitating separate lines of research. Since many Music Informatics tasks suffer from data scarcity, a mid-level representation that works for multiple tasks (i.e., a common "front-end") would be highly beneficial, allowing whatever training data is available to be fully exploited for training the task-specific part of each system (the "back-end").

Given a common multipitch representation, multiple-f_0, melody, bass, and vocal f_0 tracking can be formulated as nearly identical tasks: Given a set of candidates, identify the relevant subset of candidates. This formulation is easily extended to different sub-tasks, such as applying different definitions of melody [8] or selecting the pitch of a specific type of source (e.g. female vocals).

Just as phonemes are the building blocks of speech, pitch contours—continuous trajectories of fundamental frequency values over time, whose length may vary from a single note in the shortest case to a short phrase in the longest [1, 22]—are the building blocks of pitched musical audio. They are compact and semantically meaningful units of sound organization that are well aligned with human perception of pitch in the auditory streaming literature [4], and can robustly serve as mid-level representations for pitch estimation and characterization tasks such as melody extraction. Unlike piano-roll style "notes", contours carry a rich set of information about the character of a pitched sound since they explicitly encode frequency deviations, shapes such as bends, vibrato, and melisma are captured, as well as the amplitude envelope.

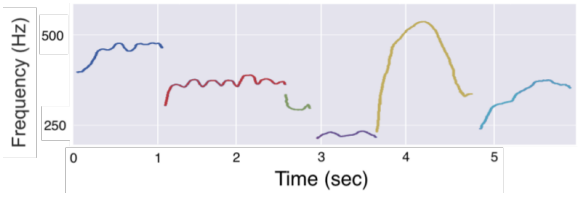

Fig. 2: Examples of six contours for an excerpt of female singing, each a different color. Salience $s(t)$ (not pictured) would be in the z-axis.

More formally, a contour is a time series $c(t) = (f(t), s(t))$ defined along a discrete, finite time interval $\{t_0, t_1, ..., t_n\}$, where $f(t)$ is the fundamental frequency of the contour over time, and $s(t)$ is a representation of the "salience" (or loudness, dominance) of the contour over time. Note that a given contour is only defined over the time interval $[t_0, t_n]$. To enforce the concept of continuity over time and pitch, contours are defined to have a minimum length of τ ms and a maximum frequency change of δ cents per ms. These values can be varied depending on the task/application. Instances of $f(t)$ for six contours are illustrated in Figure 2.

The pitch content in musical audio can be represented as a set of contours, as we have seen in Figure 1, and sets of contours can be used as a mid-level representation for various pitch estimation tasks in Music Informatics.

3 How are Contours Estimated?

To date, a number of tasks have relied explicitly on the use of contours, or "f_0 groups," including melody estimation [1, 2, 12, 17, 19, 21, 23], bass tracking [2, 7, 14, 24], singing or playing-style similarity and classification [25, 26, 27], and emotion classification [28].

One way contours are extracted is by first estimating a "pitch salience function"—a time-frequency representation that estimates the dominance or likelihood of the pitches present in the signal. The audio or spectral transform can optionally be first pre-processed to emphasize pitched content using harmonic-percussive source separation [29], spectral whitening [30, 3], a bandpass filter [2] or an equal loudness filter [18]. Salience

functions have been computed using harmonic summation across frequency [2, 3, 18, 30, 31], source-filter modeling [15, 21], a combined source-filter and harmonic summation model [17, 20], sinusoid-noise modeling [32], non-negative matrix factorization [33, 34] or through learned models [29, 35, 36, 37]. Contours are most often extracted from a pitch salience function by greedy peak streaming [17, 18, 23], where contours are created by tracing trajectories using a set of rules to enforce continuity over time, pitch and amplitude. Another approach to contour tracking is a time-domain approach [22] based on a *harmonic locked loop* (HLL), which is a frequency-locked-loop modified to jointly track harmonics.

To gain an understanding of how much of the pitched content of a music recording these methods cover, we compared the contours extracted by these methods to the set of corresponding manual pitch annotations (the "reference") and computed the fraction of references pitches that are covered by the extracted contours, for three datasets: Bach10 [31] (multiple-f_0), MedleyDB [8] (melody), and Orchset [20] (melody). The challenge of maximizing coverage (contour recall) while maintaining a high enough precision to make the representation useful (a trivial solution for maximizing recall would be to mark every possible pitch as active all the time, which is of course useless) remains open. The results are shown in Table 1.

We see that even on Bach10, a relatively simple multiple-f_0 dataset, the best method does not yet reach 100% contour coverage. MedleyDB contains more complicated polyphonic mixtures, but in this scenario the algorithms were only evaluated on their coverage of the main melody line. We see that the best method only covers 70% of the main melody, and on Orchset only 58% of the main melody is covered by the extracted contours. This lack of full pitch contour coverage highlights an important shortcoming in pitch estimation tasks that is often overlooked, because systems are being evaluated based on their final output only, without examining the mid-level pitch representations. It also highlights the need for continued research in contour estimation, where there is still substantial room for improvement.

Method	Bach10	MedleyDB	Orchset
Bosch [17]	-	0.62	0.58
Salamon [18]	-	0.64	0.45
HLL [22]	0.75	0.70	-
Duan [31]	0.73	-	-
Benetos [33]	0.90	-	-

Table 1: Average coverage of different contour tracking methods on three datasets. MedleyDB scores are computed for melody type 2 [8].

4 What Can Contours Represent?

Given an estimated contour, there are a number of semantically meaningful features that can be extracted from it, such as basic statistics and polynomial models of the pitch and salience (mean, standard deviation, range), duration information, and template matching [38, 26, 28]. Descriptors about the vibrato can also be estimated, such as the vibrato rate, amplitude, and "coverage" (the percentage of a contour with vibrato) [1, 26]. Feature learning on contours is yet to be explored, with all previous work using hand-designed features and features from the time-series analysis literature.

To evaluate the usefulness of contours as a representation in a way which is as independent as possible from the quality of contour estimation from polyphonic music, we make use of a multitrack dataset. In this way, contours can be estimated more cleanly from the isolated monophonic recording of each instrument. The data used in the subsequent experiments was drawn from the MedleyDB multitrack dataset (first and second release) [8, 39] and the Bach10 dataset.

For each song in the corpus that was recorded in an isolated environment, we started by estimating the fundamental frequency curve for each isolated stem (track) of a monophonic instrument[2] using the pYIN pitch tracking algorithm [40]. We then filtered the fundamental frequency time-series using instrument activation confidence values (see [8]), such that if the confidence was below 0.5, any estimated pitch content was removed[3]. Contours were created from the filtered fundamental frequency curves by segmenting the curve into contiguously active regions. Regions were further segmented if the change in pitch was greater than $\delta = 13.8$ cents per ms (80 cents per 5.8 ms hop), and contours shorter than $\tau = 25$ ms were removed [4]. Pitch salience is estimated as the weighted sum of the spectral energy at the

[2] Polyphonic instruments such as piano were ignored.

[3] The parameters were optimized to maximize the filtered pYIN performance against human annotations.

[4] The values of τ and δ were empirically determined based on initial experiments.

Feature	Q1	Q2 (Median)	Q3
Pitch Mean (Hz)	82.2	176.9	296.2
Pitch Slope (cents / s)	-180.2	-3.4	160.8
Pitch Std (cents)	8.7	20.6	45.7
Pitch Range (cents)	37.5	88.4	175.9
Duration (s)	0.13	0.26	0.52

Table 2: Quartile boundaries for a subset of the contour features across the dataset.

first 8 harmonics of the contour f_0, computed from the song's mix. The resulting dataset contained 109,294 contours across 28 instrument classes estimated from 529 isolated stems within 221 different songs. Features were computed for each contour as in [26]. This feature set includes polynomial fit coefficients, residuals, and basic statistics (range and standard deviation) for both frequency and salience, as well as measures of vibrato.

We measured the accuracy of our resulting pitch contours by comparing them against the 96 human-labeled annotations in MedleyDB (original release), and found they had a raw pitch accuracy [5] of 80% with an overall accuracy [6] of 79%. Most of the mistakes made by the pitch tracker were neither octave nor voicing mistakes, indicating that most of the errors are noise in the estimation. While these contours are not perfect estimates of the true pitch, we will see that they are discriminable nonetheless.

4.1 Are Contours Organized in Feature Space?

Given these cleanly computed contours and their features, we first explore how different classes of contours are organized in feature space. Table 2 shows the quartile boundaries for a subset of the features across the set of contours. We randomly select approximately 6700 pitch contours from the 10 most common instruments in our dataset (including vocals), and project them into a 2-dimensional space using t-Distributed Stochastic Neighbor Embedding (t-SNE) [41], a technique designed for visualizing and revealing structure in high-dimensional data. In our case, the original dimensions are given by the relatively small set of contour features described previously. The result is displayed in Figure 3, where in each subplot we highlight a different subset of instruments: (a) string (violin, viola and cello), (b) bass (double and electric), (c) vocals (male and female, including rap) and (d) wind (saxophone and trumpet). We see that even with a relatively small set of features derived directly from the pitch contours, different instrument classes tend to cluster in different regions of the embedding. This suggests that pitch-contour-based features are semantically rich and discriminative despite their low dimensionality. This makes them an attractive mid-level representation, which could potentially be used to discriminate between instruments or relevant groups of instruments such as the vocals or the bass line. In the following section we explore this further through a series of supervised learning experiments.

4.2 Can we Train a Model to Discriminate Between Contours?

In order to directly quantify the discriminability of contours, we consider the following binary classification tasks: (1) vocal / non-vocal, (2) bass / non-bass, (3) melody / non-melody, and (4) male singer / female singer (on the subset of vocal contours). For each task, a binary 100-tree random-forest classifier is trained on the features described earlier in Section 4. The features were standardized to zero mean and unit variance, and the parameters of the classifiers were fit using a randomized hyper-parameter search on the training set. The training and test set were created using an artist-conditional random split on the songs in the dataset (see [19]), treating all tracks from Bach10 as a single artist. Each experiment was run on five different random splits, and the class-weighted accuracy is shown in Figure 4, top.

The bass classification tasks performs best at 94% average accuracy, followed by melody and vocals. While there is still room for improvement, these results demonstrate that even with a basic feature set, if the contour estimates are relatively clean, tasks such as melody and bass line selection can be performed at the contour level. In analyzing the importance of each of the features, we found that the classifier relied heavily on "average pitch". To better understand the influence of this feature, we trained a second set of models on the same feature set, removing average pitch. The results are shown in Figure 4, bottom. For bass, we see that the accuracy drops by ≈20% without the use of the average pitch feature, and similarly the melody and vocal classification accuracies drop by ≈15%, indicating that the other features have enough discriminative information

[5] The percentage of voiced frames correct within a quarter tone.
[6] The percentage of correct voiced and unvoiced frames.

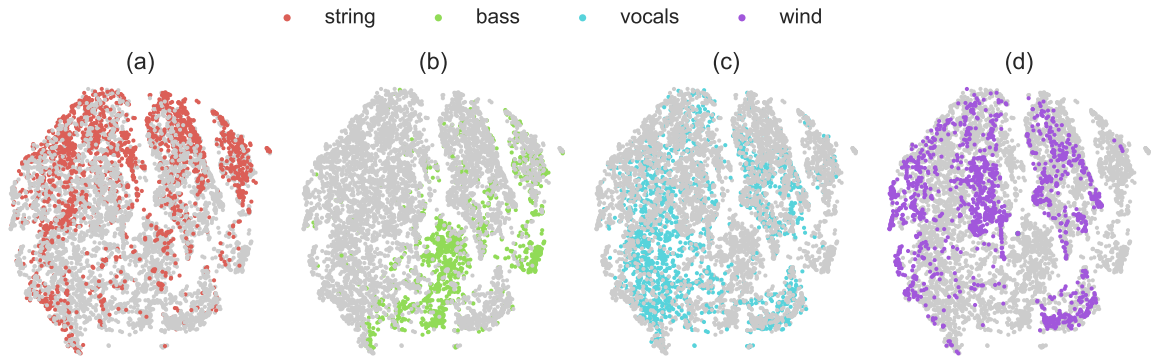

Fig. 3: t-SNE projection of 6700 contours from 10 instruments grouped into 4 categories: (a) string (violin, viola & cello), (b) bass (double & electric), (c) vocals (male & female) and (d) wind (saxophone & trumpet).

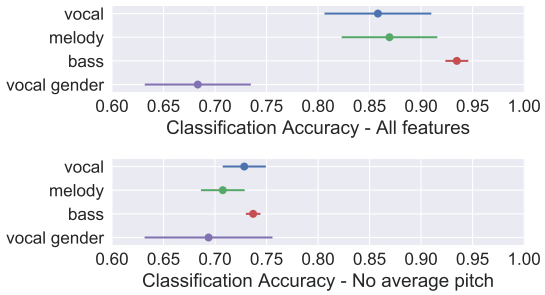

Fig. 4: Class-weighted classification accuracy for each task when training with all features (top) and without average pitch (bottom). Bars indicate the standard deviation across experiments.

to classify the majority of contours. Interestingly, we see that vocal gender is not at all influenced by removing average pitch, likely due to the often overlapping vocal range of male and female singers in our dataset.

4.3 Does this approach also work for polyphonic mixes?

We have shown that even with a basic feature set, contours belonging to different classes can be discriminated when the contours are cleanly estimated from isolated signals. This raises the question of how well contours estimated from polyphonic audio can be discriminated.

In Melodia [1], contours are estimated using the peak-streaming approach described in Section 3, and are discriminated based on a set of heuristic rules about their features. Using the same set of contours computed by the Melodia front end, we explored using a generative statistical model for distinguishing melody from non-melody contours [38] and found that the performance nearly matched the performance of Melodia on a per-contour basis. We also tried using a discriminative classifier and found that it outperformed the generative model, correctly classifying 75% of contours vs. 72% of contours [19]. We repeated the same experiment using contours computed from an improved salience function, evaluating on both the MedleyDB and Orchset datasets [17], and found that the improved salience function resulted in more discriminative contour features. Using supervised learning to train a model for contour classification (compared to a fixed set of heuristics) was especially useful in the case of orchestral music, since rules from Melodia had been tailored to other types of data. We performed a similar experiment to distinguish vocal from non-vocal contours in a database of world folk music [26] using a discriminative classifier and achieved a class-weighted per-contour accuracy of 74%.

We also used the Melodia-based contours to explore the usefulness of melodic contour features for singing/melodic style classification [25]. In Figure 5 we reproduce a plot from this study that illustrates the potential discriminative power of contour features: using only two vibrato-related features, we see that the five melodic styles considered in the study are already fairly separable.

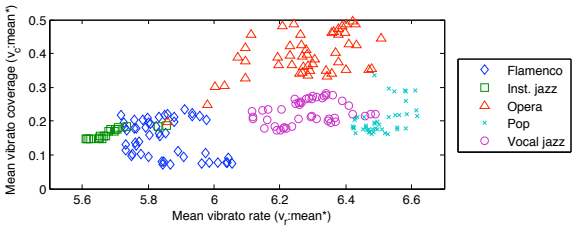

Fig. 5: Singing style as a function of mean vibrato rate and coverage. Reproduced from [25] with permission from the authors.

5 Future Directions

We have seen that contour coverage and estimation quality is a major bottleneck in pitch-estimation tasks, with none of the approaches presented to date reaching 100% contour recall. Given the promise that deep learning approaches have shown for melody extraction and other tasks in Music Informatics, we see an opportunity in this domain. Deep learning could be used to learn a time-frequency representation akin to a multipitch salience function that would facilitate better contour tracking. Since this multipitch representation can be shared by multiple tasks, there is a clear opportunity for applying multitask learning to jointly learn a multipitch representation. Additionally, incorporating phase information into contour extraction methods could help untangle overlapping or interfering partials.

We applied a relatively simple feature set in our experiments, and it is almost certainly undercomplete. This could be addressed by feature learning techniques, as well as by adding more information to a contour, such as a representation of the amplitudes of harmonics over time as a proxy for timbre. Contour classifiers to date use less information than the heuristic methods—namely they use little information about neighboring or co-occurring contours. These relationships could potentially be accounted for in a decoding stage, much like a language model in speech recognition. Additionally, including timbre information (e.g. the amplitude of each harmonic over time), would likely improve contour discrimination tasks such as vocal gender classification.

To facilitate research on contour-driven Music Informatics, as well as the reproducibility of the experiments presented in this study, we have created an open source library called `motif` [7] built around the factorization paradigm we have proposed in this paper. The library contains implementations of several contour extraction and contour classification methods that can be applied to any pitch estimation task. The library is built to make it easy to add new methods and experiment with combinations, and we encourage contributions.

References

[1] Salamon, J. and Gómez, E., "Melody Extraction from Polyphonic Music Signals using Pitch Contour Characteristics," *IEEE TASLP*, 20(6), pp. 1759–1770, 2012.

[2] Goto, M., "A Real-Time Music-Scene-Description System: Predominant-F0 Estimation for Detecting Melody and Bass Lines in Real-World Audio Signals," *Speech Communication*, 43(4), pp. 311–329, 2004.

[3] Klapuri, A., "Multiple Fundamental Frequency Estimation based on Harmonicity and Spectral Smoothness," *IEEE TASLP*, 11(6), pp. 804–816, 2003.

[4] Bregman, A. S., *Auditory scene analysis*, MIT Press, Cambridge, Massachussetts, 1990.

[5] Cho, T., Weiss, R. J., and Bello, J. P., "Exploring common variations in state of the art chord recognition systems," in *SMC*, pp. 1–8, 2010.

[6] Humphrey, E. J. and Bello, J. P., "From music audio to chord tablature: Teaching deep convolutional networks to play guitar," in *ICASSP*, pp. 6974–6978, 2014.

[7] Salamon, J., Serrà, J., and Gómez, E., "Tonal representations for music retrieval: from version identification to query-by-humming," *IJMIR, special issue on Hybrid Music Info. Retrieval*, 2(1), pp. 45–58, 2013.

[8] Bittner, R. M., Salamon, J., Tierney, M., Mauch, M., Cannam, C., and Bello, J. P., "MedleyDB: A Multitrack Dataset for Annotation-Intensive MIR Research," in *ISMIR*, 2014.

[9] Amodei, D., Anubhai, R., Battenberg, E., Case, C., Casper, J., Catanzaro, B., Chen, J., Chrzanowski, M., Coates, A., Diamos, G., et al., "Deep speech 2: End-to-end speech recognition in english and mandarin," *arXiv:1512.02595*, 2015.

[10] Chigier, B., "Automatic speech recognition," 1997, uS Patent 5,638,487.

[11] Cancela, P., López, E., and Rocamora, M., "Fan chirp transform for music representation," in *DAFx*, 2010.

[12] Paiva, R. P., Mendes, T., and Cardoso, A., "Melody Detection in Polyphonic Musical Signals: Exploiting Perceptual Rules, Note Salience, and Melodic Smoothness," *Comput. Music J.*, 30(4), pp. 80–98, 2006.

[7] `http://www.github.com/rabitt/motif`

[13] Dressler, K., "Pitch Estimation by the Pair-Wise Evaluation of Spectral Peaks," in *AES Conference*, Audio Engineering Society, 2011.

[14] Ryynanen, M. and Klapuri, A., "Automatic bass line transcription from streaming polyphonic audio," in *ICASSP*, volume 4, pp. IV–1437, IEEE, 2007.

[15] Durrieu, J.-L., David, B., and Richard, G., "A musically motivated mid-level representation for pitch estimation and musical audio source separation," *IEEE J. on Selected Topics on Signal Processing*, 5(6), pp. 1180–1191, 2011.

[16] Cancela, P., "Tracking Melody in Polyphonic Audio," in *4th Music Inform. Retrieval Evaluation eXchange (MIREX)*, 2008.

[17] Bosch, J., Bittner, R. M., Salamon, J., and Gómez, E., "A Comparison of Melody Extraction Methods Based on Source-Filter Modeling," in *ISMIR*, pp. 571–577, New York, 2016.

[18] Salamon, J., Gómez, E., and Bonada, J., "Sinusoid Extraction and Salience Function Design for Predominant Melody Estimation," in *DAFx-11*, pp. 73–80, Paris, France, 2011.

[19] Bittner, R. M., Salamon, J., Essid, S., and Bello, J. P., "Melody Extraction by Contour Classification," in *ISMIR*, 2015.

[20] Bosch, J. J., Marxer, R., and Gómez, E., "Evaluation and combination of pitch estimation methods for melody extraction in symphonic classical music," *JNMR*, pp. 1–17, 2016.

[21] Bosch, J. and Gómez, E., "Melody extraction based on a source-filter model using pitch contour selection," in *SMC*, pp. 67–74, Hamburg, Germany, 2016.

[22] Bittner, R. M., Wang, A., and Bello, J. P., "Pitch Contour Tracking in Music Using Harmonic Locked Loops," in *ICASSP*, New Orleans, USA, 2017.

[23] Dressler, K., "An Auditory Streaming Approach for Melody Extraction from Polyphonic Music," in *ISMIR*, 2011.

[24] Hainsworth, S. W. and Macleod, M. D., "Automatic Bass Line Transcription from Polyphonic Music." in *ICMC*, 2001.

[25] Salamon, J., Rocha, B., and Gómez, E., "Musical Genre Classification using Melody Features Extracted from Polyphonic Music Signals," in *ICASSP*, pp. 81–84, Kyoto, Japan, 2012.

[26] Panteli, M., Bittner, R. M., Bello, J. P., and Dixon, S., "Towards the Characterization of Singing Styles in World Music," in *ICASSP*, New Orleans, USA, 2017.

[27] Abeßer, J., Frieler, K., Cano, E., Pfleiderer, M., and Zaddach, W. G., "Score-Informed Analysis of Tuning, Intonation, Pitch Modulation, and Dynamics in Jazz Solos," *IEEE/ACM TASLP*, 25(1), pp. 168–177, 2017.

[28] Panda, R., Rocha, B., and Paiva, R. P., "Dimensional music emotion recognition: Combining standard and melodic audio features," in *CMMR*, pp. 583–593, 2013.

[29] Rigaud, F. and Radenen, M., "Singing Voice Melody Transcription using Deep Neural Networks," in *ISMIR*, pp. 737–743, 2016.

[30] Ryynänen, M. and Klapuri, A., "Automatic Transcription of Melody, Bass Line, and Chords in Polyphonic Music," *Computer Music J.*, 32(3), pp. 72–86, 2008.

[31] Duan, Z., Pardo, B., and Zhang, C., "Multiple fundamental frequency estimation by modeling spectral peaks and non-peak regions," *IEEE TASLP*, 18(8), pp. 2121–2133, 2010.

[32] Yeh, C., Roebel, A., and Rodet, X., "Multiple fundamental frequency estimation and polyphony inference of polyphonic music signals," *IEEE TASLP*, 18(6), pp. 1116–1126, 2010.

[33] Benetos, E. and Weyde, T., "An efficient temporally-constrained probabilistic model for multiple-instrument music transcription," in *ISMIR*, pp. 701–707, 2015.

[34] Fuentes, B., Badeau, R., and Richard, G., "Harmonic Adaptive Latent Component Analysis of Audio and Application to Music Transcription," *IEEE TASLP*, 21(9), pp. 1854–1866, 2013.

[35] Kum, S., Oh, C., and Nam, J., "Melody Extraction on Vocal Segments Using Multi-Column Deep Neural Networks," in *ISMIR*, ISMIR, 2016.

[36] Poliner, G. and Ellis, D., "A Classification Approach to Melody Transcription," in *ISMIR*, pp. 161–166, London, 2005.

[37] Verma, P. and Schafer, R. W., "Frequency Estimation from Waveforms Using Multi-Layered Neural Networks," *Interspeech*, pp. 2165–2169, 2016.

[38] Salamon, J., Peeters, G., and Röbel, A., "Statistical Characterisation of Melodic Pitch Contours and its Application for Melody Extraction," in *ISMIR*, pp. 187–192, Porto, Portugal, 2012.

[39] Bittner, R. M., Wilkins, J., Yip, H., and Bello, J. P., "MedleyDB 2.0: New Data and a System for Sustainable Data Collection," in *ISMIR Late Breaking and Demo Papers*, 2016.

[40] Mauch, M. and Dixon, S., "PYIN: a Fundamental Frequency Estimator Using Probabilistic Threshold Distributions," in *ICASSP*, pp. 659–663, IEEE, 2014.

[41] van der Maaten, L. and Hinton, G., "Visualizing data using t-SNE," *JMLR*, 9(Nov), pp. 2579–2605, 2008.