VOCAL MELODY EXTRACTION VIA DNN-BASED PITCH ESTIMATION AND SALIENCE-BASED PITCH REFINEMENT

Yongwei Gao[1], *Bilei Zhu*[1], *Wei Li*[1,2], *Ke Li*[3], *Yongjian Wu*[3], *Feiyue Huang*[3]

[1] School of Computer Science, Fudan University, China
[2] Shanghai Key Laboratory of Intelligent Information Processing, Fudan University, China
[3] Tencent Youtu Lab, China

ABSTRACT

Data-driven methods for melody extraction from polyphonic music generally require large amounts of labeled data for model training. However, musical data with annotations of melody fundamental frequency (F0) are rare and hard to obtain. To overcome this limitation, in this paper we propose to use melody MIDI files, which are more massively available, as the sources of labels to train a deep neural network (DNN) model for melody extraction. For each testing audio, the pitch sequence estimated by DNN is comprised of note numbers quantized at semitone level, and their resolution is relatively low. Therefore, we further propose a salience-based method to refine the pitch estimate of DNN to a higher resolution of 10 cents. Experimental results on three public datasets indicate that our method outperforms four state-of-the-art melody extraction methods in most cases.

Index Terms— Melody extraction, melody MIDI, pitch resolution, deep neural network (DNN), salience-based method

1. INTRODUCTION

Melody extraction is the process of automatically obtaining a sequence of fundamental frequency (F0) that represents the pitch of the dominant melodic line of a polyphonic music piece. This is an important task in music information retrieval (MIR), with numerous potential applications such as query by humming, cover song identification and singing voice separation [1].

Existing algorithms for automatic melody extraction can be generally classified into three categories: salience-based methods (e.g., [2, 3, 4, 5]), source separation-based methods (e.g., [6, 7]) and data-driven methods (e.g., [8, 9, 10, 11, 12]). In particular, with the popularity of deep learning, data-driven methods based on deep neural networks are gaining more and more attentions in the research of melody extraction. It is well known that deep learning models generally require massive data for training. However, resources of precise annotations of melody F0 are rare, and thus the size of data available to train a melody classification model is usually small. This is one of the major reasons that prevent the end-to-end models from achieving significantly better performance than salience-based and source separation-based methods [9]. To address this issue, different strategies such as data augmentation [9], semi-automatic dataset generation [11], and NMF-based salience representation [13] have been tried.

Data augmentation and dataset generation both aim at obtaining a larger and well-labeled dataset. Unlike them, in this paper we propose a new solution to solve the small data size problem faced when using deep learning for melody extraction. Specifically, our solution is based on the observation that compared with the rare datasets annotated with precise F0, melody MIDI files, which encode the note information of each music recording's melody, are more easily accessible. By decoding these melody MIDI files, a sequence of pitch values representing the melody of each corresponding music recording can be obtained, and these pitch sequences can be used as labels to train deep learning models of melody extraction.

Unfortunately, pitch values extracted from MIDI files are quantized at semitone level (1 semitone = 100 cents) and the resolution is relatively low. Moreover, pitch fluctuation within each note is ignored. As a result, models trained using melody MIDI files can only provide semitone-level pitch estimation for each testing audio. However, in many MIR applications such as singing voice separation, the precision of pitch estimation is essential for the overall performance [14], and in other applications such as vocal and instrumental activity recognition, using pitch fluctuation as a feature can significantly boost the recognition accuracy [15]. In these cases, an estimation of melody at a higher resolution is needed.

In spite of the low resolution, we argue that the semitone-level pitch sequence can facilitate the calculation of melody F0 at the level of cents, as it provides a pitch trend to narrow the search range of the target F0. To verify this argument, in this paper we propose a two-stage method for melody extraction of polyphonic music. In the first stage, a DNN model is trained using 22,000 polyphonic music recordings of various genres and their corresponding melody MIDI files to classify the frames of each testing audio to obtain a sequence of semitone-level pitch values (i.e., note numbers). In the second

stage, the pitch sequence obtained is refined to 10-cent level via a salience-based approach. Experiments on three public datasets show that our method outperforms four state-of-the-art algorithms significantly in most cases.

2. ALGORITHM DESCRIPTION

As demonstrated in Fig. 1, our algorithm contains two stages. The first stage uses a DNN model trained on melody MIDI files to extract a sequence of semitone-level pitch values from each input audio. The second stage refines this pitch sequence to a resolution of 10 cents based on a salience-based method.

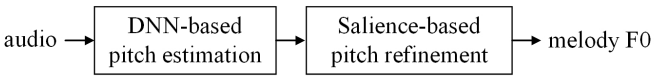

Fig. 1. Framework of our algorithm.

2.1. Semitone-Level Pitch Estimation via DNN

In our previous work [16], we presented a DNN model trained on 2,246 music recordings and their corresponding melody MIDI files, which achieved significantly better performance than the *melodia* method [4] in estimating the melody at semitone level from polyphonic music. In the current study, a similar DNN model is utilized for semitone-level pitch estimation.

Our DNN model contains an input layer, three hidden layers and an output layer, classifying the melody of the input audio frame by frame. The input layer takes as input the features derived from Constant-Q transform (CQT) [17] of the input audio. The CQT is calculated with a hop length of 23.2 ms, and for each frame the feature vector is formed by stacking the current CQT spectrum and those of the 20 preceding frames and the 20 succeeding frames. The three hidden layers are fully connected, each of which has 1,024 hidden units and uses the rectified linear unit (ReLU) for activation. The output layer uses the *softmax* function to obtain the posterior distribution of each pitch class against a total of 61 classes (corresponding to frequencies from 55 Hz to 1.76 kHz and an "unvoiced" class).

To train the DNN model, 22,000 music recordings and their corresponding melody MIDI files are used (in contrast, the model in [16] was trained on 2,246 songs). These recordings are all vocal-accompaniment-mixed popular songs and each melody MIDI file is manually transcribed from the corresponding audio in house by music editors. During the training, the standard stochastic gradient descent (SGD) algorithm is used to minimize the cross entropy loss function.

Given a testing music recording, the trained DNN model produces a pitch sequence consisting of note numbers, which is then smoothed by median filtering with a window size of 27. Each pitch value in the obtained pitch sequence, although with relatively low resolution, gives us an approximate estimation near which we can find the melody F0 of the corresponding time frame with a higher resolution. The search of a more-precise F0 at each frame is performed by using a salience-based method, as described in the next section.

2.2. Salience-Based Pitch Refinement

As concluded in [1], a majority of existing melody extraction algorithms are salience-based. These algorithms first calculate a salience function to measure the salience of each possible F0 value over time, and then at each frame they examine all the peaks of this salience function to identify the peak corresponding to the melody. Our pitch refinement approach follows this procedure, except that we already have a pitch estimate (i.e., the semitone-level pitch estimate given by the DNN model) for each frame and we only have to examine the peaks near this pitch estimate.

We adopt the method in [4] to construct a salience function for each music recording. First, short-time Fourier transform (STFT) is applied with the window length and the hop length set to 46.4 ms and 23.2 ms respectively. Instantaneous frequency is then calculated to obtain a more accurate estimate of the peak's frequency and amplitude. Spectral peaks are then selected by finding local maxima at each frame of the magnitude spectrogram.

The salience function we use covers the frequency range from 55 Hz to 1.76 kHz, quantized into 600 bins where each bin contains 10 cents. For each frame, the salience value at the bin b is calculated as the weighted sum of the spectral peaks detected at this frame,

$$S(b) = \sum_{h=1}^{N_h} \sum_{i=1}^{I} e(a_i) \times g(b, h, f_i) \times (a_i)^{\beta}, \qquad (1)$$

where N_h is the number of harmonics considered, I is the number of peaks found, a_i and f_i are the amplitude and frequency of the ith peak respectively, $e(a_i)$ is a magnitude threshold function, $g(b, h, f_i)$ is a weighting function, and β is an amplitude compression parameter. For details of calculating the salience function, please refer to [4].

After obtaining the salience function, we then utilize it to refine our semitone-level pitch estimate. This is done frame by frame. For the pitch value at frame t in the sequence, we first convert it from note number N_t to Hertz f_t, and assign it to one of the 600 bins b_t in the salience function, as follows,

$$f_t = 2^{(N_t - 69)/12} \times 440, \qquad (2)$$

$$b_t = \lfloor \frac{1200 \times \log_2(f_t/55)}{10} + 1 \rfloor. \qquad (3)$$

For each frame, the bin b_t gives an approximate estimation where the target F0 value with 10-cent resolution lies.

Then at each frame of the salience function we examine the bin b_t calculated from Eq. (3), as well as the K bins upwards and the K bins downwards. The bin b_t^{max} with maximal salience value within the $2K+1$ bins is considered to contain the refined melody F0 and thus selected out. Finally, b_t^{max} is converted to a Hertz value as follows,

$$f_t^{refined} = 2^{(b_t^{max}-1)\times 10/1200} \times 55, \quad (4)$$

where $f_t^{refined}$ is the refined melody F0 at the frame considered.

In this paper, the voicing decision is determined by the outputs of DNN plus median filtering, we do not explore more sophisticate decision method.

3. EXPERIMENTS

3.1. Datasets and Evaluation Metrics

The performance of our method was evaluated on three public datasets: ADC2004[1], MIREX05[1] and MedleyDB [18]. The free music datasets of RWC (J-RWC) [19] was used as validation dataset for parameter selection. Music recordings in all these four datasets were sampled at 44.1 kHz. Since the training set of our DNN model consists of mostly vocal-accompaniment-mixed songs, only recordings with a lead voice in these datasets were used for testing.

To measure the performance of melody extraction, five metrics were calculated by using the *mir_eval* toolbox [20]: voicing recall rate (VR), voicing false alarm rate (VFA), raw pitch accuracy (RPA), raw chroma accuracy (RCA) and overall accuracy (OA). On each dataset considered, we calculated the mean of each metric over all music recordings in the dataset. Detailed descriptions of these metrics can be found in [1]. In our experiments, we followed the settings in [4] and considered an estimated pitch value to be correct when its absolute difference with the ground truth is less than 50 cents.

3.2. Experimental Results

3.2.1. Parameter Selection

K is a parameter used for pitch refinement in Section 2.2. At each frame of the salience function, it determines the range of search for the target frequency bin (where the high-resolution F0 lies) given the frequency bin determined by Eq. (3). In this experiment, we varied K from 0 to 40 with a step of 5, and for each value of K we ran our method on the J-RWC dataset to calculate the mean OA, mean RPA, and mean RCA. As shown in Fig. 2, when K increases, each metric increases monotonically with K first, reaches their maxima at $K = 25$, and then declines slowly. This is as expected. When $K \leq 25$, a wider search range brings larger chance to find the target frequency bin. However, when $K > 25$, there may be too many interfering options, which makes it hard to find the target frequency bin. As a result, we fixed the value of K to 25 in all the following experiments.

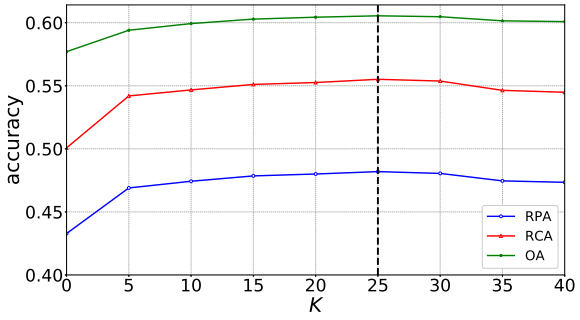

Fig. 2. The effect of K on the performance of melody extraction on the J-RWC dataset.

3.2.2. The Effect of Pitch Refinement

We have shown in [16] that our proposed DNN model is powerful in estimating the semitone-level pitch for musical audio. In the proposed study, we conducted a new set of experiments to verify the effectiveness of the other stage, i.e., pitch refinement, of our melody extraction algorithm. These experiments were carried out on the three public testing datasets, and for each audio in these datasets, we first converted its pitch annotations from Hertz to note numbers via

$$N = 12 \times \log_2(f/440) + 69, \quad (5)$$

where f is the frequency value in Hertz, and N is the corresponding note number, to obtain the ground-truth pitch sequence at the semitone level. This sequence was then fed to our pitch refinement method to produce the final pitch estimation.

By using the ground-truth semitone-level pitch as input instead of the values given by our DNN model, we can eliminate the effect of the DNN model and focus on evaluating the performance of pitch refinement. The mean RPA, RCA and OA for each of the three datasets obtained from the above strategy (denoted by *semiGT + PR*) are illustrated in Table 1. For comparison, we also give in this table the mean RPA, RCA and OA of directly converting the ground-truth note numbers to Hertz using Eq. (2) (denoted by *semiGT*). It can be seen from Table 1 that on all the three datasets, the metrics of *semiGT* are poor. This is obviously due to the precision loss produced by Eq. (5). Fortunately, *semiGT+PR* can undo the loss to a high extent, and the mean OA values are all above 91%. This clearly indicates the effectiveness of our salience-based pitch refinement method.

Fig. 3 gives an example of pitch refinement for a short segment in the ADC2004 dataset. The upper subplot shows

[1] https://labrosa.ee.columbia.edu/projects/melody/

in the salience function the semitone-level melody line estimated by the DNN model, and the bottom subplot illustrates the refined pitch sequence and the ground-truth melody line. As we can see from this figure, our pitch refinement method works well in recovering pitch fluctuation.

Table 1. The effect of the salience-based pitch refinement.

Datasets	Algorithms	RPA	RCA	OA
ADC2004	semiGT	46.9	46.9	55.3
(vocal)	semiGT+PR	90.3	90.3	92.2
MIREX05	semiGT	54.1	54.1	70.2
(vocal)	semiGT+PR	86.3	86.4	91.5
MedleyDB	semiGT	50.1	50.1	77.2
(vocal)	semiGT+PR	89.6	89.6	96.5

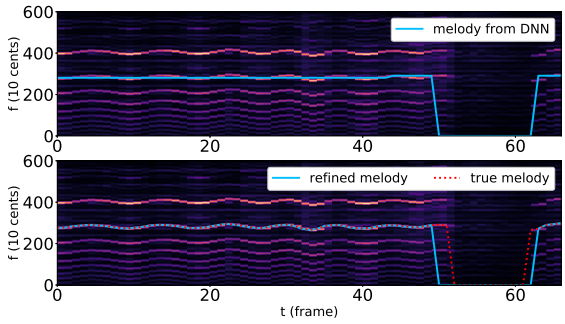

Fig. 3. Pitch refinement for a segment of "opera_male5 .wav" in the ADC2004 dataset.

3.2.3. System Evaluation

To assess the performance of our system, two strategies of our method were evaluated: 1) *DNN*, which directly converts the semitone-level pitch estimate given by DNN to Hertz values via Eq. (2); 2) *DNN+PR*, which represents our full system consisting of DNN-based pitch estimation and salience-based pitch refinement. For comparison, four state-of-art melody extraction methods, i.e., *melodia*, *MCDNN* [9], *DeepSalience* [11], and *patch-CNN* [12] were also evaluated. Please note that *MCDNN* and *DeepSalience* were not evaluated on the MedleyDB dataset, since this dataset was used in their training. All the comparative methods were run at their default settings.

Our experimental results on ADC2004, MIREX05 and MedleyDB are listed in Table 2, 3 and 4 respectively. As we can see from these tables, *DNN+PR* achieved significantly higher RPA, RCA and OA than *DNN* in all cases. This indicates again the effectiveness of our salience-based pitch refinement method. Compared with the four state-of-the-art methods, *DNN+PR* achieved the best overall performance in terms of OA on ADC2004 and MedleyDB. Moreover, it performed better than all competitors in terms of RPA. Measured at RCA, *DNN+RP* won most of the competitions on ADC2004 and MedleyDB, except that it achieved slightly lower RCA than *melodia* on the MedleyDB dataset.

From Table 3 we can see that *DNN+PR* achieved a relatively lower OA than *melodia*, *DeepSalience* and *patch-CNN* on the MIREX05 dataset. This is mainly due to the poor performance of our DNN model in discriminating vocal and nonvocal segments. As we can see in the table, *DNN* and *DNN+PR* achieved a VFA of 42.4%, which is significantly higher than *melodia* and *DeepSalience*.

Table 2. ADC2004 (vocal)

Algorithms	OA	RPA	RCA	VR	VFA
melodia	73.9	71.8	74.8	81.6	**12.0**
MCDNN	73.1	75.8	78.3	88.9	41.2
DeepSalience	72.3	71.2	74.8	76.2	13.7
patch-CNN	72.3	74.8	76.1	**90.5**	42.1
DNN	62.0	62.1	63.4	87.3	28.6
DNN+PR	**75.0**	**77.5**	**79.2**	87.3	28.6

Table 3. MIREX05 (vocal)

Algorithms	OA	RPA	RCA	VR	VFA
melodia	76.6	76.7	77.9	87.0	22.7
MCDNN	68.4	76.3	77.4	87.0	49.0
DeepSalience	**79.9**	73.9	74.7	79.6	**10.0**
patch-CNN	73.3	**82.9**	**83.5**	**96.0**	43.9
DNN	56.3	55.5	56.1	89.2	42.4
DNN+PR	70.3	76.6	77.2	89.2	42.4

Table 4. MedleyDB (vocal)

Algorithms	OA	RPA	RCA	VR	VFA
melodia	61.9	63.3	**72.3**	**84.6**	38.9
patch-CNN	61.1	58.9	63.7	75.8	40.4
DNN	56.6	45.6	50.6	79.1	**35.9**
DNN+PR	**65.4**	**63.4**	69.6	79.1	**35.9**

4. CONLUSION

In this paper, we present a two-stage method for melody extraction from polyphonic music. The first stage performs pitch estimation based on a DNN model trained using 22K audio files and their corresponding melody MIDI files. Then the estimated pitch values are refined in the second stage via a salience-based pitch refinement method. Experiments on three datasets show that the performance of our method is significantly better or at least comparable to the four state-of-the-art methods.

For future work, we are currently trying to improve the performance of voicing detection for our algorithm, which is the one of the main weaknesses of our proposed algorithm. This can be done by, for example, joining audio features with vocal characteristics.

5. ACKNOWLEDGEMENT

This work was supported by NSFC 61671156.

6. REFERENCES

[1] 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*, vol. 31, no. 2, pp. 118–134, 2014.

[2] M. Goto, "A real-time music-scene-description system: Predominant-f0 estimation for detecting melody and bass lines in real-world audio signals," *Speech Communication*, vol. 43, no. 4, pp. 311–329, 2004.

[3] M. P. Ryynänen and A. P. Klapuri, "Automatic transcription of melody, bass line, and chords in polyphonic music," *Computer Music Journal*, vol. 32, no. 3, pp. 72–86, 2008.

[4] J. Salamon and E. Gómez, "Melody extraction from polyphonic music signals using pitch contour characteristics," *IEEE Trans. on Audio, Speech, and Language Processing*, vol. 20, no. 6, pp. 1759–1770, 2012.

[5] V. Rao and P. Rao, "Vocal melody extraction in the presence of pitched accompaniment in polyphonic music," *IEEE Trans. on Audio, Speech, and Language Processing*, vol. 18, no. 8, pp. 2145–2154, 2010.

[6] J. L. Durrieu, G. Richard, B. David, and C. Févotte, "Source/filter model for unsupervised main melody extraction from polyphonic audio signals," *IEEE Trans. on Audio, Speech, and Language Processing*, vol. 18, no. 3, pp. 564–575, 2010.

[7] H. Tachibana, T. Ono, N. Ono, and S. Sagayama, "Melody line estimation in homophonic music audio signals based on temporal-variability of melodic source," in *Proc. of the IEEE Int. Conf. on Acoustics, Speech and Signal Processing (ICASSP)*, Texas, USA, 2010, pp. 425–428.

[8] G. E. Poliner and D. P. W. Ellis, "A classification approach to melody transcription," in *Proc. of the Int. Society for Musical Information Retrieval Conf. (ISMIR)*, London, UK, 2005, pp. 161–166.

[9] S. Kum, C. Oh, and J. Nam, "Melody extraction on vocal segments using multi-column deep neural networks," in *Proc. of the Int. Society for Musical Information Retrieval Conf. (ISMIR)*, New York, USA, 2016, pp. 819–825.

[10] F. Rigaud and M. Radenen, "Singing voice melody transcription using deep neural networks," in *Proc. of the Int. Society for Musical Information Retrieval Conf. (ISMIR)*, New York, USA, 2016, pp. 737–743.

[11] R. M. Bittner, B. McFee, J. Salamon, P. Li, and J. P. Bello, "Deep salience representations for f0 estimation in polyphonic music," in *Proc. of the Int. Society for Musical Information Retrieval Conf. (ISMIR)*, Suzhou, China, 2017, pp. 23–27.

[12] L. Su, "Vocal melody extraction using patch-based cnn," in *Proc. of the IEEE Int. Conf. on Acoustics, Speech and Signal Processing (ICASSP)*, Alberta, Canada, 2018.

[13] B. Dogac, E. Slim, and P. Geoffroy, "Main melody extraction with source-filter nmf and crnn," in *Proc. of the Int. Society for Musical Information Retrieval Conf. (ISMIR)*, Paris, France, 2018, pp. 82–89.

[14] B. Zhu, W. Li, and L. Li, "Towards solving the bottleneck of pitch-based singing voice separation," in *Proc. of the ACM Int. Conf. on Multimedia (MM)*, Brisbane, Australia, 2015, pp. 511–520.

[15] M. Mauch, H. Fujihara, K. Yoshii, and M. Goto, "Timbre and melody features for the recognition of vocal activity and instrumental solos in polyphonic music," in *Proc. of the Int. Society for Musical Information Retrieval Conf. (ISMIR)*, Miami, Florida, 2011, pp. 233–238.

[16] B. Zhu, F. Wu, K. Li, Y. Wu, F. Huang, and Y. Wu, "Fusing transcription results from polyphonic and monophonic audio for singing melody transcription in polyphonic music," in *Proc. of the IEEE Int. Conf. on Acoustics, Speech and Signal Processing (ICASSP)*, New Orleans, USA, 2017, pp. 296–300.

[17] J. C. Brown, "Calculation of a constant q spectral transform," *The Journal of the Acoustical Society of America*, vol. 89, no. 1, pp. 425–434, 1991.

[18] R. Bittner, J. Salamon, M. Tierney, M. Mauch, C. Cannam, and J. Bello, "Medleydb: A multitrack dataset for annotation-intensive mir research," in *Proc. of the Int. Society for Musical Information Retrieval Conf. (ISMIR)*, Taipei, Taiwan, 2014, pp. 155–160.

[19] M. Goto, H. Hashiguchi, T. Nishimura, and R. Oka, "Rwc music database: Popular, classical and jazz music databases," in *Proc. of the Int. Society for Musical Information Retrieval Conf. (ISMIR)*, Paris, France, 2002, pp. 287–288.

[20] 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 *Proc. of the Int. Society for Musical Information Retrieval Conf. (ISMIR)*, Taipei, Taiwan, 2014.